UNIVERSITY OF LIVERPOOL

FRUGALITY IN SET-SYSTEM AUCTIONS

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Antony McCabe

July 2012

Contents

Notations ix

Preface xi

Abstract xiii

Acknowledgements xv

1 Introduction 1
- 1.1 History . . . 1
  - 1.1.1 Auction Mechanisms . . . 3
  - 1.1.2 Example of a Vickrey Auction . . . 3
    - Truthfulness of Vickrey Auction . . . 4
    - Notation . . . 4
- 1.2 Procurement Auctions . . . 5
  - 1.2.1 Set-System Auctions . . . 6
- 1.3 Truthful Strategies . . . 7
  - 1.3.1 The Vickrey-Clarke-Groves (VCG) Mechanism . . . 7
  - 1.3.2 VCG Example . . . 8
- 1.4 Frugality . . . 9
  - 1.4.1 Frugality Definitions . . . 11
  - 1.4.2 Benchmarks . . . 12
  - 1.4.3 Feasible Bid Vectors and Nash Equilibrium . . . 13
- 1.5 Special Cases of Set-Systems . . . 14
  - 1.5.1 Path Auctions . . . 14
  - 1.5.2 Commodity Auctions . . . 14
- 1.6 Thesis Outline . . . 15
  - 1.6.1 Chapter 2 . . . 15
  - 1.6.2 Chapter 3 . . . 15
  - 1.6.3 Chapter 4 . . . 15
  - 1.6.4 Chapter 5 . . . 16
  - 1.6.5 Chapter 6 . . . 17
  - 1.6.6 Chapter 7 . . . 17

2 Frugality in General Set-System Auctions 19
- 2.1 Introduction . . . 19

2.2 Frugality of VCG 20
2.3 Frugality of Approximation Algorithms 21
2.3.1 Considering the Minimal Winning Sets 22

3 The Single-Commodity Auction **25**
3.1 Overview 25
3.2 Definitions 26
3.3 The $\{1, 2\}$ Single-Commodity Auction 26
3.3.1 Auction Definition 26
3.3.2 The $\alpha\mathcal{M}$ Mechanism 27
3.3.3 Computing NTUmin for the {1,2} Single-Commodity Auction . . 28
Preliminaries 29
Describing the NTUmin bids. 29
Computing the NTUmin bid values 37
Simplifying lower bounds for NTUmin 39
3.4 The $\alpha\mathcal{M}$ mechanism 41
3.4.1 Frugality results for the $\alpha\mathcal{M}$ mechanism 41
3.4.2 A lower bound on frugality with the $\alpha\mathcal{M}$ mechanism 48
Calculation of a lower bound 50
3.5 The Unrestricted Integer Single-Commodity Auction 52
3.5.1 A lower bound on frugality for all Scaling Mechanisms 52
Preliminaries 53
Proof of lower bound 53

4 Shortest Path with k-sets **61**
4.1 Overview 61
4.2 Problem Definitions and Examples 62
4.2.1 Problem Definitions 63
4.2.2 Example 65
4.3 Hardness results 65
4.3.1 SHORTEST PATH WITH 3-SETS 66
4.3.2 SHORTEST PATH WITH 2-SETS 67
4.4 Auction Design 72
4.4.1 Using VCG 72
4.4.2 Mechanism $\mathcal{M}^{\mathcal{P}}$ 72
4.4.3 Frugality Results for $\mathcal{M}^{\mathcal{P}}$ 73
Lower bounds for $\mathcal{M}^{\mathcal{P}}$ 75
4.4.4 Inapproximability Results 76

5 Benchmarks and First-Price Auctions **79**
5.1 Overview 79
5.2 Hypergraph Representation of Constraints 81
5.3 Descending Price Auction 84
5.4 Uniformly Descending Price Auction 88
5.5 Ascending Price Auction 90
5.6 Ascending from Zero Auction 92
5.7 Uniformly Ascending Auctions 93

5.8 Path Auctions 98
5.9 Ordered Maximal Bidding 101
5.9.1 Definitions 102
5.9.2 Examples 103
Random Orderings 104
Bounds for Values Achievable with Ordered Bidding 105
5.9.3 Restricted Setting 110
Single-Commodity Auctions 113
5.9.4 Results for $|S| \leq 4$ 117
5.9.5 Upper Bound for Restricted Settings 122
5.9.6 Hardness and Approximation Results 124

6 Benchmarks for Forward Auctions **131**
6.1 Overview 131
6.2 Definitions 131
6.3 Comparison with Combinatorial Auctions 132
6.3.1 Definitions 132
6.3.2 Comparison of Set-System Auctions and Combinatorial Auctions . 133
6.4 Benchmarks 134
6.4.1 Optimal Solution as a Benchmark 134
6.4.2 Considering F1NTUmax as a Benchmark 135
Randomized Mechanisms 136
Non-Optimal choices of winning set 137
6.4.3 Considering F2NTUmax as a Benchmark 138
6.4.4 Considering FNTUmax as a Benchmark 138
6.4.5 Benchmarks for Unit Demand Auctions 139
6.4.6 Considering Alternatives to FNTUmax 140

7 Conclusion and Discussion **143**
7.1 Conclusion 143
7.1.1 Discussion and Summary of Main Results 143

Bibliography **147**

Illustrations

List of Figures

4.1 Example of M3C . . . 65
4.2 Figure 4.1 as SHORTEST PATH WITH 3-SETS . . . 65
4.3 Construction for V_i' to V_{i+1}' . . . 68
4.4 Example of SHORTEST PATH WITH 3-SETS . . . 68
4.5 Figure 4.4 as SHORTEST PATH WITH 2-SETS . . . 68
4.6 Construction of Example for Lower Bound of $\mathcal{M}^{\mathcal{P}}$ . . . 76

5.1 Hypergraph Representation of Constraints for Table 5.1 . . . 84
5.2 Uniformly Ascending Price Auction for Shortest Path Auction . . . 99
5.3 A Shortest Path Auction Where the Uniformly Ascending from Zero Price is Below NTUmin . . . 100
5.4 A Shortest Path Auction where the Uniformly Ascending from Zero Price is Above NTUmax . . . 101
5.5 Hypergraph Representation of constraints for Table 5.11 . . . 106
5.6 Hypergraph Representation of Constraints for Table 5.12 . . . 107
5.7 Hypergraph Representation of Constraints for Table 5.13 . . . 109
5.8 Graph Representation of Constraints in Table 5.14 . . . 111
5.9 Graph Representation of Constraints in Table 5.15 . . . 113

List of Tables

1.1 Mary's Vickrey Auction . . . 5
1.2 Mary's VCG Auction . . . 9
1.3 Mary's VCG Auction with NTUmin, NTUmax . . . 13

3.1 Instance I with VCG ($\alpha = 1$) . . . 28
3.2 Instance I with $\alpha = Q$ . . . 49
3.3 Instance I' with $\alpha = Q$ . . . 50
3.4 Instance I' with $\alpha = 1$ (equivalent to VCG) . . . 50
3.5 Instance I with $\alpha\mathcal{M}$ . . . 52
3.6 Instance I' with $\alpha\mathcal{M}$ . . . 52
3.7 Instance I^1 . . . 54
3.8 Instance I^2 . . . 55
3.9 Instance I^j . . . 56
3.10 Instance I^k . . . 58

5.1 Showing Constraint Sets for a Set-System . . . 83
5.2 Descending Price Auction Reaches 0 . . . 86

5.3 Descending Price Auction as low as TUmin/2 89
5.4 Uniformly Ascending Bid $b^{\uparrow}$ Equals NTUmax 95
5.5 Uniformly Ascending Bid $b^{\uparrow}$ equals NTUmin 96
5.6 Example 5.4 Multiplied by constant k 97
5.7 Example 5.6 substituting A_1 with $A'_1, \ldots, A'_k$ 98
5.8 Ordered Maximal Bid Auction for 9 items giving differing results 103
5.9 Ordered Maximal Bid Auction, choosing randomly 105
5.10 Ordered Maximal Bid Auction may not find NTUmax 105
5.11 Ordered Maximal Bid Auction may not find NTUmin when $|S| = 5$ 106
5.12 Ordered Maximal Bid Auction May Not Find NTUmin 107
5.13 Ordered Maximal Bid Auction May Be Much Higher than NTUmin 109
5.14 Ordered Maximal Bid Auction may not equal NTUmin in restricted setting . 111
5.15 OMBmin approaches 2 NTUmin 113
5.16 OMBmin approaches 2 NTUmin for Single-Commodity Auctions 114
5.17 Ordered Maximal Bid Auction, Lowest Cardinality First 116
5.18 Ordered Maximal Bid Auction, Highest Cardinality First 116
5.19 Ordered Maximal Bid Auction, Lowest Quantity First 117

List of Algorithms

1 Algorithm to calculate NTUmin bids 38

2 Approximation Algorithm for SPk 73

3 A Class of Descending Price Auctions 85
4 A Uniformly Descending Price Auction 89
5 A Class of Ascending Price Auctions 90
6 A Class of Ascending from Zero Auctions 92
7 A Uniformly Ascending Price Auction 94

List of Problems

1 SHORTEST PATH WITH k-SETS 63
2 EXACT COVER BY 3-SETS 63
3 MINIMUM SET COVER 64
4 MINIMUM k-SET COVER 64
5 MINIMUM INDEPENDENT DOMINATING SET 125
6 MINIMUM WEIGHT EDGE COVER 128

Notations

The following notations and abbreviations are found throughout this thesis:

$\mathbb{Z}$	The set of integers.
$\mathbb{Z}^+$	The set of strictly positive integers.
$\mathcal{E}$	The set of participating agents ($\mathcal{E} = \{1, \ldots, n\}$).
n	The number of participating agents ($n = \lvert\mathcal{E}\rvert$).
$\mathcal{F}$	The collection of feasible sets.
S	The lowest cost feasible set.
$S^{\mathcal{M}}$	The feasible set chosen by mechanism $\mathcal{M}$.
m	The number of agents in the feasible set (e.g. $m = \lvert S\rvert$).
$\mathbf{b}$	A bid vector ($\mathbf{b} = (b_1, \ldots, b_m)$).
b_e	The bid of agent e.
$\mathbf{b}^\alpha$	A specific bid vector ($\mathbf{b}^\alpha = (b_1^\alpha, \ldots, b_m^\alpha)$).
b_e^α	The bid of agent e in bid vector $\mathbf{b}^\alpha$.
b_V	The sum of bids of agents in set $V (\sum_{e \in V} b_e)$.
c_V	The sum of costs of agents in set $V (\sum_{e \in V} c_e)$.
b_V^α	The sum of bids, in vector $\mathbf{b}^\alpha$, of agents in set $V (\sum_{e \in V} b_e^\alpha)$.
$p_V^{\mathcal{M}}$	The sum of payments by $\mathcal{M}$ to agents in set $V (\sum_{e \in V} p_e^{\mathcal{M}})$.
$p^{\mathcal{M}}$	The sum of payments by $\mathcal{M}$ to agents in the set $S^{\mathcal{M}} (\sum_{e \in S^{\mathcal{M}}} p_e^{\mathcal{M}})$.
$\mathbf{c}$	A cost vector ($\mathbf{c} = (c_1, \ldots, c_n)$).
c_e	The cost of agent e.
$\mathbf{p}^{\mathcal{M}}$	The payment vector of mechanism $\mathcal{M}$, ($\mathbf{p} = (p_1, \ldots, p_m)$).
$p_e^{\mathcal{M}}$	The payment to agent e by the mechanism $\mathcal{M}$.
NTUmin	Non-Transferable Utility minimum.
$\mathbf{b}^{\min}$	An NTUmin bid vector
NTUmax	Non-Transferable Utility maximum.
TUmin	Transferable Utility minimum.

TUmax	Transferable Utility maximum.
OMBmin	Ordered Maximal Bidding minimum.
OMBmax	Ordered Maximal Bidding maximum.
σ_S	An ordering of the set S.
σ	An ordering of the lexicographically first, lowest cost feasible set, S.
$\mathbf{b}^\sigma$	A maximal ordered bid vector for ordering σ.
σ^γ	A specific ordering of the lexicographically first, lowest cost feasible set, S.
$\mathbf{b}^\gamma$	A maximal ordered bid vector for ordering γ.
$\mathbf{b}^{\Uparrow,f}$	An iteratively rising bid vector for selection function f.
$\mathbf{b}^{\Downarrow,f}$	An iteratively rising bid vector for selection function f.
$\mathbf{b}^\uparrow$	A uniformly rising bid vector.
$\mathbf{b}^\downarrow$	A uniformly falling bid vector.
Q	A quantity to purchase in a single-commodity auction.
$\mathbf{q}$	A quantity vector ($\mathbf{q} = (q_1, \ldots, q_n)$).

Preface

This thesis is primarily my own work. The sources of other materials are identifed. Chapter 1 contains introductory materials describing the setting, and is drawn from various authors. From Chapter 2 onwards, of particular importance are the definitions and notation that have been used in previous literature describing set-system auctions and frugality. These come from the paper by Elkind, Goldberg and Goldberg in 2007 [11], which derives much of the notation and definitions from Karlin, Kempe and Tamir in 2005 [24]. The structure for various examples are derived from examples given in [11]; in particular extensions of the 'diamond' graph appearing in both papers.

Abstract

We study the topic Frugality in Set System auctions; examining the payments that are given by truthful mechanisms when buying selections of items at auction.

Firstly, we examine a simple single-commodity auction, where the auctioneer wishes to buy some given quantity of identical items. We show methods of quickly computing a winning set, as well as the benchmark NTUmin. We then show, for certain special cases, a mechanism that improves on the frugality of VCG, and is within a constant factor of optimal for mechanisms in its class. We then consider the general case, and see a relatively large lower-bound for a class of similar mechanisms.

We propose a new type of auction, based on finding the shortest-path in a graph with 'bundles' of edges. We show that finding an optimal solution to this problem is NP-hard, for any bundle-size (k) of 2 or more, showing that there is no polynomial time algorithm that can compute an exact solution, subject to the commonly-held assumption that P! =NP. However, we give a simple k approximation and use this to design a truthful mechanism and give its frugality ratio.

We consider the benchmarks that have been used in the literature as first-price auctions, and examine a range of other possibilities that should aim to meet the same 'fairness' criteria. We show that not all of the proposals will meet these criteria, and give the ranges of values possible for these other benchmarks. We also give information on their computational complexity, including a new result showing approximation hardness for a new benchmark as well as an existing one used in the literature.

We then briefly examine the meaning of the benchmarks we used for frugality if they are rewritten for use in the more traditional 'forward' auctions (that is, selling items by auction).

Acknowledgements

I would like to thank my supervisors Paul Goldberg and Leslie Ann Goldberg for suggesting this line of research and the support I have received throughout my PhD studies. It is only Leslie's intervention that resulted in my starting a PhD, or even knowing what it was.

I would also like to thank all the members of the University of Liverpool Computer Science Department, including the excellent technical and support staff for making this a pleasant place to work. I am indebted to the department, and the EPSRC, for the funding that made this possible. I have had many excellent colleagues in my office and building over this time, who have all been helpful, interesting and supportive.

Particular thanks are due to my examiners, David Manlove and Giorgios Christodoulou for their attention and constructive comments.

Finally, I would like to thank my family for helping me to concentrate on my studies and forget about other distractions, and my friends for doing the opposite!

Chapter 1

Introduction

1.1 History

An auction is a traditional method of selling items, or services, amongst a pool of interested parties. In the classic setting, an item is offered up for sale and 'bids' are invited by varying means, with the intention that the seller can agree to sell to one of the interested parties at some price acceptable to the chosen buyer. Reports of sales by auction go back at least as far as Herodotus [21], around 2,500 years ago.

One of the most common types of auction is an 'English' auction, which is an interactive process, traditionally held with all of the potential buyers in the same room. In this auction, the auctioneer starts by offering the item at some low price. When at least one person indicates their willingness to buy at the current price, they are deemed to have *bid* that price on the item, and are accepted, provisionally, as a winner at the price of the bid. The auctioneer then increases the price, and asks if anyone else in the room would like to buy at the slightly higher price. If there is somebody else who is prepared to pay the new price, they then become the provisional winner at the new price. This process is then repeated until there are no bidders left that are prepared to pay the new price, and the provisional winner becomes the winner and must pay the price of his final bid. Although it is not relevant in this work, for completeness, it is worth describing a reasonably common complication — a seller may specify a *reserve* price, which is not communicated to the buyers, and if the result of the auction would be to sell below this price then the sale is cancelled.

Another type of auction that has been commonly used is the *sealed bid* auction. In such an auction, the seller invites the buyers to submit a bid amount, in a sealed envelope, to be given by a certain date and time. At the given time no more bids are accepted, and the sealed bids can now be revealed. The seller then chooses the amount of the largest bid and sells to that bidder at the price of his bid (this property is often called a *first-price* auction, as the price paid is equal to the highest bid).

William Vickrey in 1961 [37] made some interesting observations about the different strategies that are present in these two types of auction. Assuming that all the buyers have in mind a specific valuation for the item (the maximum price that they are prepared to pay) which only they know, their optimal strategy in an English auction is to keep bidding until the price reaches their private valuation.

The optimal strategy for a sealed bid is rather more complex, as it will depend on the buyers' knowledge of the likely bids by the other buyers. As an example, if each buyer were to believe that his is the only bid, he could place an arbitrarily low bid with the expectation that he could secure the purchase at a very low price. There is also certainly no incentive to place a bid that is in excess of their valuations, so this auction tends to incentivize bidding below their valuation, but not in an easily predicted way. If we wish to make any sort of mathematical analysis of an auction then being unable to determine the correct strategy for any of the buyers is obviously undesirable.

Vickrey proposed a sealed-bid auction that avoids this problem. His auction takes sealed bids, chooses the winner with the highest bid, but only sells at the price of the next highest bid. He shows, through case analysis, that the optimal strategy in this type of auction is for each of the sellers to submit a bid which is equal to their full valuation for the item. This type of auction is now generally known as a *Vickrey Auction*, and the term *second-price* is often used as a description. We will see an example of this shortly, in Section 1.1.2, which will also describe the case analysis which shows that each bidder should only ever bid their full, truthful, valuation.

A *dominant strategy* in an auction is some strategy that a buyer can adopt in order to guarantee their best possible outcome, no matter what the bids of the other buyers may be. When the dominant strategy for a type of auction is truth-telling, we often describe it as *truthful*. Obviously, not all types of auction admit a dominant strategy, such as the 'first-price' sealed-bid auction discussed earlier — in order to get the best possible result, the bidder would need to know all of the other bids first.

While, of course, humans are quite capable of employing many convoluted, devious and even unfathomable strategies, the notion of the 'rational, selfish agent' is perhaps not so far from reality. As Vickrey pointed out, truthfulness removes the (possibly substantial) burden for each bidder of trying to predict the behaviour of the other bidders. If this difficult prediction is not done correctly then it seems likely that this will lead to less than optimal outcomes.

The notion of truthfulness is not restricted to auctions, and can be applied to many other settings which involve making some allocation of resources within some community, such as in auctions, markets, voting and routing games. Perhaps, unsurprisingly, there is a great deal of literature studying *truthful mechanisms* and there are many references as well as descriptions of several domains in the book by Nisan et al. [35].

So far, we have been discussing sales by auction as they have been historically the most commonly seen type of auction. However, this thesis is primarily about *procurement* auctions where the auction is run on behalf of a single buyer, rather than a seller. Sales by auction can sometimes be called *forward* auctions with procurement auctions being called *reverse* auctions. We primarily concern ourselves with sealed bid auctions (although we do not restrict ourselves to a single item), and the notion of *truthfulness* in these auctions is an important theme in this thesis.

As is common with the literature in the field, throughout this work we refer to the auction participants as *agents*, with the body deemed to be running the auction called the *auctioneer.*

One assumption we make, which is extremely natural in an auction setting, is that we imagine the use of some form of currency or money. The way that we imagine the use of money throughout this work is to assign values to certain properties that the participants hold. We measure how 'happy' an agent is with some *utility* value, so our assumption that each participant acts purely selfishly (or *rationally*) can be modelled by requiring that every agent maximizes its utility value.

1.1.1 Auction Mechanisms

We generally call these rules concerning the running of an auction a *mechanism* (see, e.g., [3, 4, 24, 35, 11]).

In the case of a sealed-bid procurement auction a mechanism may be uniquely characterized by just two rules — a *selection* rule and a *payment* rule. The selection rule examines all of the bids supplied by the agents and then chooses which of the agents it will buy from. The payment rule will then decide a value for a payment to each of the agents that was selected. We will also make the standard, reasonable, assumptions on *individual rationality* — that an agent which is not selected will always have a payment value of zero, and that each agent that is selling an item will be paid no less than the amount that they bid, otherwise they would not be willing to participate in the auction.

The same concept of a mechanism can be applied to a forward auction. The mechanism will select a 'winning set' to sell items to, and will decide on the payment amount that each of the winning agents must make. It is also standard to assume that agents who do not win pay zero, and that no agent is asked to pay more than the amount of their bid.

1.1.2 Example of a Vickrey Auction

We now give a more detailed description of a procurement auction. In this example, we have a person, Mary who wishes to purchase a pet lamb. Mary has identified a number of possible suppliers. The first on her list is a nearby farm owned by Fred. Fred would be able to get £20 each for his lambs at the local market, but making a special delivery to Mary would cost him an additional £5 in time and fuel, so Fred values his lamb at £25. The second supplier on Mary's list is George's Farm, which is somewhat further away. Including the delivery, George values his lamb at £40. The final possible supplier is a neighbour and small-holder, Bob. He also keeps sheep as pets and has had more lambs this year than he expected; he would be happy to let one go to a good home (and not to slaughter) for free to reduce his maintenance costs, and hence he values his lamb at zero, although he would still like to be paid a fair price.

Mary has chosen to purchase her lamb by way of a sealed-bid auction, and has chosen to use the Vickrey, 2nd-price, auction. In this auction, Mary will ask for sealed-bids and will select the lowest-cost supplier. She will then pay the second-lowest cost. As Fred, George and Bob are all aware of this, they can each see that their best strategy is simply to bid their actual valuations, of £25, £40 and 0 respectively. As Vickrey noted, there is never any advantage to bidding other than their valuations in this setting, but we will briefly demonstrate it here.

Truthfulness of Vickrey Auction

We will examine Fred's possible choice of bids, assuming that the other bids are unknown to him. We take the first case, that Fred bids £25 and wins the auction. This means that Fred is paid the second-lowest price, £x and that $x \geq 25$. If Fred bids below £25 he will still win the auction, and still be paid £x so does not benefit by changing it. If Fred bids above £25, and still wins the auction, then he must have bid no more than x, so is still paid £x. If Fred bids above £25, but loses the auction, then he makes no profit at all, so cannot benefit by this.

In the second case, Fred bids £25 but loses in the auction, so somebody must have bid £x, where $x \leq 25$. If he bids more than £25, he will still lose. If he were to bid £y with $y < 25$ and he manages to win the auction, then he must have bid $y \leq x$, but $x \leq 25$, so he cannot make any profit with this bid either (and will make a loss unless $x = 25$ exactly). This analysis may be duplicated for each of the other suppliers with the obvious changes.

Notation

Now we have seen why the Vickrey auction ensures that the suppliers bid truthfully, we will consider a more formal notation for this auction. We denote Fred by Agent 1, George by 2 and Bob by 3, and write this auction in the form given in Example 1.1; in order to avoid ambiguity we will use the labels A_1, A_2, A_3 to refer to agents $1, 2$, and 3 (this notation is used throughout this thesis).

The example shows the set of solutions that are acceptable to Mary as *feasible sets*, denoted by $\mathcal{F} = \{\{A_1\}, \{A_2\}, \{A_3\}\}$, which shows simply that Mary may have chosen to buy from any of Fred, George, or Bob. We can see from Table 1.1 in column c_e that the entry for agent A_1 gives $c_1 = 25$[1] which is the cost value for Fred, and hence the amount of the bid he submitted, as he will bid truthfully. The other costs given are $c_2 = 40$ and $c_3 = 0$.

We can see that Mary chooses to buy her lamb from Bob, where the winning set is given as $S = \{A_3\}$. The payment made to Bob of £25, as the second-highest, is given by $p_3^{\mathrm{V}} = 25$. We follow the convention that 'losing' agents do not receive any payment.

[1]The notation describing the cost of agent A_1 is abbreviated from c_{A_1} to c_1 throughout. Similar abbreviations for the bid, price and valuation of an agent A_e are given as b_e, p_e, and v_e respectively.

Example 1.1. *Suppose that we have three agents A_1, A_2 and A_3, and that the feasible sets are $\mathcal{F} = \{\{A_1\}, \{A_2\}, \{A_3\}\}$. Observe that the winning set $S = \{A_3\}$.*

Agent	c_e	p_e^V
A_1	25	
A_2	40	
A_3	0	25

TABLE 1.1: Mary's Vickrey Auction

1.2 Procurement Auctions

In recent years, competitive tendering has been of great importance to many organisations, in particular public bodies, not least due to the accountability it lends to the procurement process. For example, in the United Kingdom, The 1980 Local Government, Planning and Land Act (c.65) made competitive tendering compulsory in certain circumstances. Competitive tendering typically requires that a number of different competitors are given the opportunity to bid for a contract, with the contract awarded to the most competitive bid (typically the lowest cost, but there may be other factors taken into consideration).

The traditional auction sales that have been discussed here have often been about selling lots of single items and, similarly, 'competitive tendering' can be seen as an auction for a single contract. These may easily and naturally be extended to include auctions for multiple items, which were also studied by Vickrey [37]. From the assumption that a buyer may wish to procure some number of items it is only a small step to assume that a buyer may wish to procure only certain specific combinations of items. To give a possible example; perhaps an organisation may need promotional materials to be printed and there are a number of different technologies available. If they buy inkjet printers then they may also need plain paper and ink. Laser printing may use the same paper but require toner rather than ink. Thermal printing may require special paper but no further consumables. This choice may easily extend to many other items, such as the options for covers or bindings. It is certainly conceivable that there are many different manufacturers who may be able to competitively supply specific parts of the requirements without being able to fulfil the entire need unilaterally.

Another way of considering this type of auction is assume it that of 'hiring a team of agents' (see, e.g., [4, 24, 11]) to perform some complex task. In this model we assume that each agent can perform some task, for a price, which is a sub-task of the more complex task that is required. It is the goal of the auctioneer to buy any set of agents such that, between them, they can perform the entire complex task required.

This notion, of buying only certain combinations of items, can be captured with a set-system auction, and we will now see a more precise definition of the type of procurement auction that we will study. In the general case, we allow the acceptable solutions to be defined as a specified collection of sets, each of which is some acceptable subset of the selling agents that are taking part in the auction (which we call a feasible set). We call this auction a *Set-System Auction.*

Set-System auctions are integral to much of the work that is presented in this thesis. We will now formally define the Set-System Auction as a method of using an auction to purchase a solution to some problem that can be represented by a set-system, following the definitions of Karlin et al. [24].

1.2.1 Set-System Auctions

Let a set-system $(\mathcal{E}, \mathcal{F})$ be specified by a set $\mathcal{E}$ of n elements, each representing an agent, and a collection $\mathcal{F} \subseteq 2^{\mathcal{E}}$ of feasible sets; these are the subsets of agents that make an acceptable solution for the buyer. We will, in general, assume that any superset of a feasible set is also a feasible set, which is a reasonable assumption for many problems of 'hiring a team'.

The final outcome of the auction will be achieved by a process as follows. Each agent $e \in \mathcal{E}$ will submit a sealed bid b_e to the auction, giving a bid vector $\mathbf{b}$. Let $n = |\mathcal{E}|$, and $\mathbf{b} = (b_1, \ldots, b_n)$. We also assume that there is a cost vector $\mathbf{c} = (c_1, \ldots, c_n)$ that represents the cost c_e that each agent e will incur in providing its goods or services, but that this information is private and known only to the agent. Recall that a truthful mechanism is designed to elicit truth-telling, so that an agent may maximize its utility by truthful bidding, i.e., that $\forall e \in \mathcal{E}, b_e = c_e$. Typically, in a procurement auction, the utility value is simply defined as payment received minus cost outlaid.

In order to simplify some of the later notation, we will define aggregates over any set of agents, denoted by T, to refer to the sum of values of those agents. Recall that we have defined a cost vector $\mathbf{c} = (c_1, \ldots, c_n)$, and a bid vector $\mathbf{b} = (b_1, \ldots, b_n)$, and let $\mathbf{p}$ be a payment vector $\mathbf{p} = (p_1, \ldots, p_n)$ representing the payments made to each agent. Let these aggregates be

$$c_T = \sum_{e \in T} c_e,$$

$$b_T = \sum_{e \in T} b_e,$$

$$p_T = \sum_{e \in T} p_e.$$

Recall that a *mechanism*, which we will denote by $\mathcal{M}$, is a method (or set of rules) for running the auction. Every mechanism $\mathcal{M}$ consists of a selection rule, and a payment rule, which are implemented as follows. The auctioneer will, by means of this mechanism $\mathcal{M}$, select a set of agents (which must be a feasible set defined by the set-system) which we will call $S^{\mathcal{M}} \in \mathcal{F}$ to be the 'winners'. The mechanism will then pay each agent $e \in S^{\mathcal{M}}$ some value $p_e^{\mathcal{M}}$. Note that the set $S^{\mathcal{M}}$ will depend on the mechanism $\mathcal{M}$, and may not always be the same as the lowest-cost solution.

1.3 Truthful Strategies

Recall the idea of *dominant strategies* — which are when an auction participant (a seller or buyer, for procurement auctions or forward auctions respectively) has some specific strategy that will always give them the best possible result, regardless of the bids of other participants. We are interested in designing mechanisms that have truth-telling as a dominant strategy; that is designing auction mechanisms so that all participants are incentivized to bid only their true valuations — often referred to as a *truthful mechanism* [35].

The notion of a truthful mechanism is an interesting one, not least because it allows some method of predicting the strategies that may be employed by bidders. While this idea of truthfulness may perhaps seems a little difficult to pinpoint at first, given some reasonable assumptions, and appropriate definitions, it can be very well defined.

In a procurement auction we will generally assume that an agent is supplying some sort of goods or services and that it will incur some cost if it is selected but zero cost if it is not selected. Many of the more esoteric preferences that exist in real-world situations can be thought of as being included in this 'cost' valuation and we can think of it as the price at which the participant is ambivalent to being selected or not.

It is well-known, in our auction setting, that there are certain properties that mechanism must hold which are both necessary and sufficient for the mechanism to be truthful [35]. The first is that the selection rule must be monotonic. In a procurement auction this means that, given all other bids being equal, if agent e may be selected in the 'winning' set with some bid value b_e then agent e must always be selected with any bid b'_e when $b'_e \leq b_e$. More intuitively, an agent can never get into the winning set by increasing its bid. The second property that must be met is that the payment rule must pay *threshold payments*. A threshold payment is the supremum of the amounts that the agent can bid and still be selected in the winning set, given the fixed bids of the other agents. In the case of the Vickrey 'second-price' auctions that we have seen it is easy to observe that the threshold payment is exactly the price of the second-highest bid, as any lower bid would not win and any higher bid would be unnecessarily high.

1.3.1 The Vickrey-Clarke-Groves (VCG) Mechanism

Vickrey described a truthful way to purchase single, or multiple, items at auction. Further work, by Clarke [6] and Groves [18] then extended this mechanism to more general settings. The mechanism that they devised has become known as the Vickrey-Clarke-Groves (VCG) mechanism and is perfectly applicable to the types of set-system auctions that we are interested in. As VCG is historically important, and has a quite general definition, it is quite common for any new mechanism to be compared with VCG (see,e.g., [4, 24, 11]), and we will see a number of comparisons to the VCG mechanism throughout this work.

While the VCG mechanism is applicable to a large range of scenarios it does have a simple definition when applied to set-system auctions. We give a definition here that was described by Karlin et al. [24].

Definition 1.1. Given a set-system auction having cost vector $\mathbf{c}$, let S be the lowest-cost feasible set. Let the VCG payment denoted by p_e^{VCG} given to each agent $e \in S$, be $p_e^{VCG} = c_e + c_T - c_S$ when T is the lowest cost feasible set $T \in \mathcal{F}$ such that $e \notin T$.

□

VCG is well known to be a truthful mechanism (see, e.g., [35]), and we can easily verify this. The selection rule chooses the lowest-cost feasible set, which is clearly monotonic. The payment rule given is a threshold payment — assuming that all agents bid their cost value, then if e were to submit a bid $b_e = c_e + c_T - c_S$ then we would have $b_S = c_{S\setminus\{e\}} + c_e + c_T - c_S = c_S + c_T - c_S = c_T$. Hence $b_e = c_e + c_T - c_S$ is a threshold bid, as T is an alternative feasible set, and the cost of T compared to the cost of S represents the threshold between e 'winning' the auction and 'losing'. We already know that having a monotonic selection rule and threshold payments are sufficient conditions for a truthful mechanism.

Although the definition is given in terms of costs, we know that VCG is truthful, so we can assume that the bids made to the mechanism are identical to the costs, and the VCG payment can equally be given by $p_e^{VCG} = b_e + b_T - b_S$.

Observe that using VCG requires finding the lowest-cost feasible set. However, finding a lowest-cost feasible set may be an NP-hard problem, making VCG not universally useful. For example, Elkind et al. [11] describe a polynomial-time truthful mechanism for (NP-hard) vertex cover auctions by using an approximation algorithm.

1.3.2 VCG Example

To demonstrate VCG in action, we return to the example of Mary and her attempt to buy a pet lamb.

Mary has discovered that sheep are social animals, so she has decided that she would need to purchase at least 3 lambs. Bob still has a lamb available, and has found 2 other local small-holders that each have one lamb available that they would be prepared to accept no payment for (although Mary does not know this prior to the auction). Mary has also found a farmer who will give her two lambs for just the cost of delivery, at £40 for both.

Mary, again, decides to run an auction, and invites sealed bids. The VCG payment is slightly more complex than in Example 1.1, but the VCG payment to an agent can be thought of as the amount extra that would need to be paid if that agent were not involved in the auction. In this example, Mary will choose the 3 lambs from the small-holders, and each will be paid £40 — the cost that Mary would have incurred if each one, individually, did not participate (defined by $p_e^{VCG} = c_e + c_T - c_S$) — giving a total of £120.

This example is shown as Example 1.2, where agents A_1, A_2, A_3 represent the smallholders, and agent A_4 the farmer.

Example 1.2. *Suppose that we have four agents A_1, A_2, A_3, and A_4. The feasible sets are $\mathcal{F} = \{\{A_1, A_2, A_3\}, \{A_1, A_4\}, \{A_2, A_4\}, \{A_3, A_4\}\}$. The costs are given in Table 1.2, and observe that the winning set $S = \{A_1, A_2, A_3\}$.*

	Agent	c_e	p_e^{VCG}
	A_1	0	40
	A_2	0	40
	A_3	0	40
	A_4	40	
Total			120

TABLE 1.2: Mary's VCG Auction

1.4 Frugality

Recall that, in the Vickrey procurement auction, a single item is bought for a price equal to the second-lowest bid. Obviously, this may not be the optimal outcome for the auctioneer as there may have been a seller willing to sell at a much lower price (such as in Example 1.1). If we wish to make our mechanisms truthful, then we may have to pay more than perhaps we might had we somehow known all of the bidder's true valuations. Truthful mechanisms are certainly very desirable mechanisms for study, so we would naturally like to understand more about this potential downside of overpayment. Not surprisingly, the question of studying overpayment has been looked at earlier — initially by Archer and Tardos [4] in 2002. They called this concept, of measuring overpayment, *frugality*. The measuring of frugality is a common theme throughout this thesis, where we aim to extend the known results in this field. Firstly, though, we need to give a description of what exactly we mean by frugality, which is followed by a formal definition of a metric which allows the idea of frugality to be measured.

Perhaps the first question that springs to mind in measuring 'overpayment' would be 'what do we compare the payment with?'. It turns out that this is not a trivial question. A first attempt may be to consider taking the cost of the winning set as a possible benchmark, but even for the Vickrey mechanism, Archer and Tardos in [4] observed that no truthful mechanism can hope to get near this lowest price (within an arbitrary factor). To help illustrate this, if we refer to Mary's Vickrey auction in Example 1.1, we see that the lowest-cost feasible set has a cost of 0, and the Vickrey payment has no finite approximation ratio to this. If we were to assume that 0 was a reasonable payment for this auction, then there would be no incentive for Bob to bid truthfully, as a larger bid may well gain him some payment, causing him not to bid truthfully.

Next, we could possibly consider using the second lowest cost feasible set as a benchmark. In the case of an auction for a single item, this would give the same value as the Vickrey payment. This idea was addressed by Archer and Tardos in [4]. They considered the case of *path auctions*[2], and while they quickly discount the lowest-cost set as a possible benchmark for truthful mechanisms, they do consider the cost of the second-best disjoint solution (or feasible set) to be a reasonable benchmark. They observe that, in the case of a path auction, there must always be at least two disjoint feasible sets or else some agent holds a monopoly. (We do not consider auctions where some agent holds a monopoly, and must always be selected, as an auction is not a useful method of purchasing items in such situations.)

However, in our general setting, it is easy to see that disjoint feasible sets may not always exist, which we can demonstrate. Returning to Example 1.2, We can easily see that there are no disjoint solutions, as any solution must contain at least 3 lambs, leaving at most 2 others. Therefore, we need some more general approach than that of Archer and Tardos, and here we turn to the procedure suggested by Karlin, Kempe and Tamir in [24] and extended by Elkind, Goldberg and Goldberg in [11].

Karlin et al. [24] decided to look at *first-price* auctions in an attempt to define a reasonable benchmark figure. Recall that first-price auctions are where the buyer is paid exactly their bid price, and that first-price auctions do not generally admit dominant strategies, and so are not truthful. They decided that, due to the lack of dominant strategies in first-price auctions, they would consider the concept of a *Nash equilibrium.*

Generally, a *Nash equilibrium* is a set of *strategies* for the agents such that no agent may increase utility by unilaterally deviating from its current strategy. More specifically, in this auction setting, we can consider an agent's strategy to be simply the bid that it makes. Hence a Nash equilibrium is an assignment of bids to the agents such that no agent would increase utility by choosing a different bid. For completeness, a *pure Nash equilibrium* requires that each agent deterministically chooses a single strategy. There is an alternative — a *mixed Nash equilibrium*, where every agent may choose a number of different strategies, each with some probability.

Based upon this idea of a Nash equilibrium, they gave the following definition for a benchmark figure, which will be referred to throughout this thesis.

[2]In a path auction, the sellers represent edges in a graph, and the auctioneer wishes to purchase a path between two specified vertices of the graph. Path Auctions are described in more detail in Section 1.5.1.

1.4.1 Frugality Definitions

Definition 1.2. Let the cost vector be $\mathbf{c} = (c_1, \ldots, c_n)$. For an instance $I = (\mathcal{E}, \mathcal{F}, \mathbf{c})$ of a set-system auction, define S as a feasible set $S \in \mathcal{F}$ with the lowest cost, i.e., $\forall S' \in \mathcal{F}, (\sum_{e' \in S'} c_{e'}) \geq (\sum_{e \in S} c_e)$. Define NTUmin($\mathbf{c}$) as the solution to the following problem. [3]

Minimize $B = \sum_{e \in S} b_e$ subject to

(1) $b_e \geq c_e$ for all $e \in S$

(2) $\sum_{e \in S \setminus T} b_e \leq \sum_{e \in T \setminus S} c_e$ for all $T \in \mathcal{F}$

(3) for every $e \in S$, there is $T_e \in \mathcal{F}$ such that $e \notin T_e$ and $\sum_{e' \in S \setminus T_e} b_{e'} = \sum_{e' \in T_e \setminus S} c_{e'}$

□

For concreteness, we will assume that that S is the lexicographically first of the lowest cost feasible sets. It was shown in Proposition 24 of [11], that NTUmin($\mathbf{c}$) will give the same result for any legal choice of S as a lowest cost winning set. So we can, also, without loss of generality, assume that S is minimal with regard to set inclusion.

We can regard these three conditions less formally as something like 'fairness' criteria that should be met to qualify as a reasonable benchmark. Specifically, they can be thought of as "1) No agent should lose out by taking part, 2) No set of agents can bid more than a competing set, and 3) Every agent bids as much as possible, as long as 1) and 2) are not violated."

We can now define how we measure a mechanism's 'overpayment' more precisely by referring to its *frugality ratio*. This is the worst-case ratio (over all possible instances) of the ratio between the payments made by that mechanism and the chosen benchmark figure. Let $p_{\mathcal{M}}(\mathbf{c})$ be the total payment made by mechanism $\mathcal{M}$, given the cost vector $\mathbf{c}$. We will then use the NTUmin($\mathbf{c}$) value to define the *frugality ratio* of a mechanism $\mathcal{M}$ — which is the supremum of the ratio of $p_{\mathcal{M}}(\mathbf{c})$ to NTUmin($\mathbf{c}$) over all possible cost vectors (over all instances).

More formally, we will define this as

$$\phi_{\mathrm{NTUmin}}(\mathcal{M}) = \sup_{\mathbf{c}}(p_{\mathcal{M}}(\mathbf{c})/\mathrm{NTUmin}(\mathbf{c})).$$

We will now define the variations that were introduced by Elkind et al. [11]. Firstly the maximization version of NTUmin, as follows. Define NTUmax($\mathbf{c}$) as the solution to the optimization problem "Maximize B subject to conditions (1),(2), and (3)". There are also similar versions that capture the idea of *transferable utility*. This notion is that some agents could possibly, rationally, bid below their costs if they could 'privately'

[3] The idea that these values come from the agents' *bids* in a first-price auction, and also the notation b_e, are those used previously in the literature (see, e.g., [24, 11, 5]). However, we also use the same description and notation for describing the bids an agent may make to some mechanism to actually take part in an auction (again, consistent with the literature), even though the two concepts are not entirely equivalent.

arrange a transfer of utility from some of the other agents. Allowing artificially low bids from some of the agents may allow other agents (possibly more of them) to improve their bids, which can lead to a wider range of values[4]. We define this by modifying the first condition, such that each agent may bid below its cost value, but not below zero, as follows.

(1*) $b_e \geq 0$ for all $e \in S$

We now use this modified constraint in the following definitions.

Definition 1.3. Let TUmin(**c**) be the solution to the optimization problem "Minimize B subject to (1*),(2), and (3)".

Let TUmax(**c**) be the solution to the optimization problem "Maximize B subject to (1*),(2), and (3)".

We simply define the other frugality ratios described by Elkind et al. [11] in the same way as we did for NTUmin.

$$\phi_{\text{NTUmax}}(\mathcal{M}) = \sup_{\mathbf{c}}(p_{\mathcal{M}}(\mathbf{c})/\text{NTUmax}(\mathbf{c})).$$

$$\phi_{\text{TUmin}}(\mathcal{M}) = \sup_{\mathbf{c}}(p_{\mathcal{M}}(\mathbf{c})/\text{TUmin}(\mathbf{c})).$$

$$\phi_{\text{TUmax}}(\mathcal{M}) = \sup_{\mathbf{c}}(p_{\mathcal{M}}(\mathbf{c})/\text{TUmax}(\mathbf{c})).$$

It was noted by Elkind et al. [11] that TUmin may be too low to be a realistic benchmark, as it may even be lower than the cost of the winning set, and that TUmax may be too liberal to use as a benchmark, as it may be much higher than the sum of VCG payments. Therefore, we do not use TUmin or TUmax widely here, but there is some comparison of other possible benchmark figures with both TUmin and TUmax given in Chapter 5.

An agent $e \in \mathcal{E}$ may sometimes be referred to by a numerical index; in order to maintain clarity, these agents will often be labelled as $A_1, \ldots, A_n$ rather than the less specific $1, \ldots, n$. In order to specify different bid vectors, we will also describe some of them with a label, such as α. In these cases we will simply have $\mathbf{b}^\alpha = (b_1^\alpha, \ldots, b_n^\alpha)$ and call the aggregate

$$b_V^\alpha = \sum_{e \in V} b_e^\alpha.$$

1.4.2 Benchmarks

Now that we have these definitions, we can see the NTUmin and NTUmax values from Mary's VCG Auction that was given in Example 1.2. These are shown in Table 1.3, where the $b_e^{\min}$ column shows an NTUmin value of £40 and an NTUmax value of £60 is shown in the $b_e^{\max}$ column. The existence of the alternative feasible sets $T_1 = \{A_3, A_4\}$,

[4] Elkind et al. [11] showed a factor of $n-1$

$T_2 = \{A_1, A_4\}$, and $T_3 = \{A_2, A_4\}$ implies the following constraints (from condition (2) of Definition 1.2) respectively $b_1+b_2 \leq 40$, $b_2+b_3 \leq 40$, and $b_1+b_3 \leq 40$. It is easy to see, given that one of these constraints must be tight due to satisfying condition (3), that the minimum value possible is 40. It is also easy to see that the maximum value that satisfies these constraints is 60 (as adding the constraints together gives $2(b_1 + b_2 + b_3) \leq 120$). In this thesis, we will primarily focus on the use of NTUmin due to its appearance in the literature (e.g. [24, 39, 23, 11]).

	Agent	c_e	p_e^{VCG}	$b_e^{\min}$	$b_e^{\max}$
	A_1	0	40	40	20
	A_2	0	40	0	20
	A_3	0	40	0	20
	A_4	40			
Total			120	40	60

TABLE 1.3: Mary's VCG Auction with NTUmin, NTUmax

Using this frugality ratio to measure overpayment, Karlin et al. [24] gave examples that show VCG may overpay (relative to NTUmin) by a factor of $n-1$, where n is the number of agents. They observe that $\Omega(n)$ is obviously an upper bound on frugality [5] and hence that the VCG payment may be undesirably large.

1.4.3 Feasible Bid Vectors and Nash Equilibrium

We will consider any bid vector which satisfies conditions (1),(2), and (3) of Definition 1.2 as a *feasible* bid vector. It is worth observing the similarity of a bid vector to a pure Nash equilibrium — given a feasible bid vector, no agent has any incentive to deviate from its current bid (assuming that the bids of agents not in S are equal to their cost). A winning agent would lose utility by strictly decreasing its bid and would drop out of the winning set by strictly increasing its bid (reducing its utility to zero). Any losing agent that decreased its bid would then be bidding a value lower than its cost, which could only lead to it possibly receiving negative utility. Hence we can think of NTUmin as being something like the cheapest Nash equilibrium for a first-price auction and it is sometimes referred to in this way (see, e.g., [35] for an example). However, we would not consider this to be a conventional pure Nash equilibrium as it requires that certain assumptions are made, particularly to do with tie-breaking rules. For instance, even though NTUmin is clearly defined for path auctions, Immorlica et al. [22] have shown that, for first-price path auctions, pure Nash equilibria may not even exist.

[5] We show that $n-1$ is an upper bound in Chapter 2.

1.5 Special Cases of Set-Systems

It is very often interesting to examine special cases of these set-system auctions. There are many reasons for this, but an important one is that a well-chosen special case may provide a good model for common, real-world, requirements. There are two special cases that are particularly important to this thesis, they are *path auctions* and *single-commodity auctions.*

1.5.1 Path Auctions

One of these special cases is for path auctions, which was formally introduced in 1999 by Nisan and Ronen [33], and involves buying, by auction, some path between two specified vertices on a weighted graph. Measuring the payments made by a truthful mechanism was studied by Archer and Tardos in 2001 [3], which they applied to path auctions in 2002 [4]. This work was extended by Elkind, Sahai, and Steiglitz in 2004 [12]. In 2005, Karlin, Kempe and Tamir [24] proposed a more general framework for measuring frugality which could be applied to all set-system auctions. They also proposed a 'scaling' mechanism which has a worst-case overpayment (or frugality ratio) of $\Omega(\sqrt{n})$. They then show that it is within a factor of $2\sqrt{2}$ of optimal for any truthful mechanism. [6]

In a path auction, we assume an underlying weighted graph $G = (V, E)$ and represent the participation of each seller as an edge $e \in E$. The aim of the buyer is to purchase, by auction, any path between two specified vertices s and t. It is easy to imagine real-world examples that can be exactly, or closely, modelled with this case. Buying paths in a communications network or the use of transport infrastructure are natural things for a large organisation, like a government, to purchase. One desirable feature of the path auction is that an optimal solution can be computed quickly (in quadratic time, due to Dijkstra [9]), but with other types of set-system there may not even be a polynomial-time algorithm for finding a best solution. In this work, we will spend some time looking for reasonable auction solutions to some such 'hard' problems; there has been previous work that has attempted to do that for other hard problems (e.g. the vertex cover auctions in [11, 25]).

1.5.2 Commodity Auctions

Another special case that is particularly studied here, but is not seen in the literature, is of a single commodity auction. In this auction we have some parameter, which we denote by Q, of individual, identical items to purchase. Each seller e also has some amount, denoted by q_e, of these items to sell which they will only sell as an entire lot. Any subset of sellers with at least a quantity of Q between them is a feasible set. This is perhaps the most simple, yet still potentially useful, set system auction which makes it particularly worthy of study.

[6] The analysis of this mechanism has since been improved to a factor of 2 by Yan [39] and Chen et al. [5]

1.6 Thesis Outline

We now give a brief description of the aims and main results of this thesis.

1.6.1 Chapter 2

Chapter 2 gives a bound on frugality for VCG of $n-1$, improving on the observation of Karlin et al. [24] that the frugality ratio of VCG is obviously $O(n)$. We extend this to give a frugality ratio for any truthful mechanisms based on monotonic approximation algorithms (when the approximation algorithm is used as the selection rule).

1.6.2 Chapter 3

In Chapter 3 we mostly consider the special-case of the single-commodity auction when each agent has at most 2 items for sale. The aim here is to design a mechanism which performs better than VCG in terms of frugality, but does not have significantly greater time complexity. We give a formal definition for this auction and show, through examples, that the VCG mechanism can overpay by as much as a factor of $n-1$ even in this restricted setting. It is reasonably obvious that this results from VCG possibly choosing a large winning set, as this requires a large number of individual payments to be made. The mechanism of Karlin et al. [24] improves on VCG by using a 'scaling' mechanism that biases the mechanism towards choosing smaller winning sets, and hence having less agents to pay. Of course, this bias must not go too far, or else a small winning set would still be able to attract undesirably large payments. Their approach was to consider two possible disjoint solutions and bias towards the smaller. Unlike path auctions, monopoly-free single commodity auctions do not have the luxury of always having at least two disjoint solutions, but it is reasonably obvious that choosing agents with larger quantities will tend to result in smaller winning sets. It is this observation that motivates the scaling mechanism proposed in Chapter 3 which results in a frugality ratio that is significantly lower than VCG (recall that frugality ratios are always worst-case). However, we show that if we lift the restriction that sellers have at most two items then similar types of scaling mechanisms can only possibly have limited success in improving frugality.

1.6.3 Chapter 4

The challenge for the mechanism design in Chapter 4 is somewhat different. We firstly propose a problem that is a reasonable generalization of a path auction. In this model, we consider that each agent owns some collection of edges, and is willing to provide all of their edges for some fixed cost. If we think of 'off-peak' or 'surplus capacity' path auctions then we can reason that sellers may often be willing to sell access to their entire network for some fixed cost. Additional network usage may add little to any overheads, for example in transport or telecommunication networks. It is also reasonable to assume that having just a single payment may reduce administration (and hence costs) for both buyer and seller, making this model attractive. This type of path auction has been previously studied by Du, Sami and Shi [10]. However, in their model, they allow agents to incorrectly declare the ownership of the edges, which is a fundamental difference. They show negative results for this setting — that there is no truthful mechanism when only the edge costs are reported. Furthermore, other auctions of that type have been shown by Kempe et al. [23] to have a lower bound on frugality of $\Omega(2^n)$ for all *false-name-proof* mechanisms (where an agent is incentivized to give an honest valuation as well as an honest declaration of ownership). These results suggest that allowing agents to dishonestly report ownership would prove a major obstacle to finding frugal (truthful) mechanisms. In many real-world cases it is reasonable to assume that the ownership information may well be public knowledge, and hence we restrict our analysis to the cases that either the information is public, or analogously, that it will be honestly revealed.

The main result of this chapter is to show that finding the lowest-cost feasible set is not polynomial-time solvable (unless P=NP), hence there is no known way of running VCG in polynomial time. We then describe a mechanism for this that has bounded frugality. Although its frugality ratio is larger than VCG (by a factor of k, a parameter of the auction) this mechanism is significant as it may be computed in polynomial time, unlike VCG.

1.6.4 Chapter 5

Chapter 5 examines a number of alternative procedures to find reasonable benchmark figures for set-system auctions. Many of the results here can be thought of as being negative as they show reasons why these procedures do not give good benchmarks, but we do see some interesting results on the possible ranges of these values.

The question of what to use as a benchmark has been asked since the notion of frugality was introduced for path auctions in 2002 by Archer and Tardos [4]. In that setting they could use the fact that there are at least two disjoint solutions in any monopoly-free graph. However, this solution does not easily generalize to many other types of procurement auction — we have seen that often no disjoint solutions even exist.

Recall that Elkind et al. proposed a number of variations of this benchmark, (NTU-max, TUmin and TUmax). The variants TUmin and TUmax (that have a weakened rule 1, allowing transfer of utility between agents) were discounted as being too weak and too strong respectively. They also noted that NTUmin has some undesirable properties — it may be non-monotonic, in that increased competition between agents may actually cause the benchmark to rise. They also show that this may be NP-hard to compute, even where finding the minimum solutions are easily computed.

As we have already noted, the NTUmin and NTUmax values can be thought of as equilibrium values of first-price auctions. We have seen, in recent work, a move from the NTUmin benchmark ([24, 11]) to the NTUmax value ([5, 25]). However, in [11] it was shown that there may be a large difference between them (a ratio of $n-2$ exists for any $n > 2$). It seems reasonable, therefore, to ask the question 'What else could be used as a reasonable benchmark'. We examine this question in Chapter 5 by looking at several other, natural, types of first-price auction.

As we would wish any proposed benchmark to be seen as 'fair' we would like our benchmarks to satisfy the same three 'fairness' rules as NTUmin and NTUmax, and hence we are looking for values in the range between these two (seller-pessimistic and seller-optimistic) values. We imagine a number of iterative processes where agents are allowed to make bids according to some rules (such as each bid must be lower than the preceding bid). Our contribution is to examine the results of these and consider the range of values that may possibly be obtained in comparison with NTUmin and NTUmax, as well as examining the complexity of actually computing the result. We will also show that approximating NTUmin can be NP-hard, even when the minimum solutions may be easily computed.

1.6.5 Chapter 6

Forward auctions are, if anything, even more ubiquitous than procurement auctions. There has been much research in the area of *combinatorial auctions*, where buyers may bid on particular 'bundles' of the items for sale (e.g. Chapter 11 of [35]). However, defining a 'fair' value to compare with the price obtained by a truthful combinatorial auction does not seem like an easy question to answer. There are benchmark figures that are used in certain special-cases, such as the $\mathcal{F}^{(2)}$ value proposed by Goldberg et al. in 2006 [16] for digital goods auctions. As the NTUmin value is so well-defined for procurement auctions it seems an obvious goal to try and adapt it for use in forward auctions. In Chapter 6, we look at how we may attempt to define set-system auctions in the forward setting so that we can define a benchmark figure analogous to NTUmin. We take the special case of unit-demand forward auctions to give some comparison between the proposed benchmark, FNTUmax, and the $\mathcal{F}^{(2)}$ value used previously.

1.6.6 Chapter 7

We finish, in Chapter 7, with conclusions and a discussion of some of the problems that are left open by this work.

Chapter 2

Frugality in General Set-System Auctions

2.1 Introduction

In this chapter we briefly examine frugality ratios for general set-system auctions. It was observed by Karlin et al. [24] that there is an upper bound on the frugality ratio of VCG which is trivially $O(n)$. We present a proof that gives an exact upper bound of $n-1$ on the frugality ratio of VCG (Karlin et al. [24] also showed that VCG has a frugality ratio of at least $n-1$). We can then extend this proof and show an upper bound on the frugality ratio for truthful mechanisms which have selection rules that are monotonic approximation algorithms. While this result is, perhaps, not very difficult, it does not appear to have been documented elsewhere. Finally we give a proof that, for VCG, that the choice of winning set may be made to be minimal with regard to set inclusion.

As part of the analysis, we will often be considering the best possible (lowest cost) feasible set that is restricted to only a given subset of agents from $\mathcal{E}$. We will define some notation for this, let $d(V)$ be the best feasible set (that with the lowest sum of costs) using only agents in V where $V \subseteq \mathcal{E}$.

We will now see a lower bound for NTUmin($\mathbf{c}$), based upon this definition, which, informally, states that NTUmin must be at least as large as the worst-case cost of replacing one of the agents to make a feasible set without it.

Lemma 2.1. NTUmin $\geq \max_e c_{d(\mathcal{E}\setminus\{e\})}$.

Proof. If we choose an e that maximizes $c_{d(\mathcal{E}\setminus\{e\})}$, then we will firstly see e can always be chosen such that $e \in S$. (We can assume, without loss of generality, that $S \subset \mathcal{E}$ because if S does not have a monopoly then we can always choose some smaller feasible set $S' \subset S$ instead.) We will examine this as two cases;

Case 1: Suppose that $c_{d(\mathcal{E}\setminus\{e\})} > c_S$.

The set given by $d(\mathcal{E} \setminus \{e\})$ has been chosen with the minimal sum of costs, with the restriction that e is not included. Therefore S can only have a lower cost if it has benefited from including agent e, and therefore it follows that $e \in S$.

Case 2: Suppose that $c_{d(\mathcal{E}\setminus\{e\})} = c_S$.

As e is chosen to maximize $c_{d(\mathcal{E}\setminus\{e\})}$, and c_S is the minimum possible choice for any e, then it follows that all possible choices of e must have equal cost. That is, $\forall j \in \mathcal{E}, c_{d(\mathcal{E}\setminus\{j\})} = c_S$. Therefore, we can choose e to be any agent in $\mathcal{E}$, and hence we can choose e such that $e \in S$.

Now, assume an NTUmin bid vector $\mathbf{b}^{\min}$ and an agent e that maximizes $c_{d(\mathcal{E}\setminus\{e\})}$. Recall the definition that $b_V^{\min} = \sum_{i \in V} b_i^{\min}$ (the sum of the bids for set V), and for our chosen agent e and that bid vector $\mathbf{b}^{\min}$ satisfies condition (3) in Definition 1.2 we have

$$b^{\min}_{S\setminus T_e} = c_{T_e \setminus S} \tag{2.1}$$

for some $T_e \in \mathcal{F}$ and $e \notin T_e$. We know from choosing $d(\mathcal{E} \setminus \{e\})$, as a lowest cost solution without e, that

$$c_{T_e} \geq c_{d(\mathcal{E}\setminus\{e\})}. \tag{2.2}$$

Additionally, where $\mathbf{b}^{\min}$ satisfies condition (1) in Definition 1.2 then we have

$$b^{\min}_{S\cap T_e} \geq c_{S \cap T_e}$$

and adding this to Equation 2.1 gives

$$b^{\min}_{S\setminus T_e} + b^{\min}_{S\cap T_e} \geq c_{T_e\setminus S} + c_{S\cap T_e}$$

which can be simplified to

$$b^{\min}_S \geq c_{T_e}$$

and with Inequality 2.2, due to transitivity then

$$b^{\min}_S \geq c_{d(\mathcal{E}\setminus\{e\})}.$$

Therefore NTUmin$(\mathbf{c}) \geq \max_e c_{d(\mathcal{E}\setminus\{e\})}$, as claimed.

□

This lower bound for NTUmin$(\mathbf{c})$ is a useful result in analysing frugality ratios, and we will now see how it can be used to prove an upper bound on the frugality of VCG.

2.2 Frugality of VCG

For a minimal winning set S, and for each $e \in S$; then we can observe that a threshold bid (and hence payment p_e) can be upper-bounded by the bids of a replacement solution.

Recall that a threshold bid is the largest bid that an agent could have submitted to the mechanism and still be selected in a winning set. Hence, for any solution $T_e \in \mathcal{F}$ such that $e \notin T_e$, then $p_e \leq c_{T_e}$. To verify this; if e had submitted a bid b_e with $b_e > c_{T_e}$ then VCG would have chosen T_e instead (as T_e would bid truthfully, $b_{T_e} = c_{T_e}$), so any $b_e > c_{T_e}$ is too large for a threshold bid.

Lemma 2.2. $\forall e \in S, p_e \leq \text{NTUmin}(\mathbf{c})$.

Proof. We have defined $d(\mathcal{E} \setminus \{e\})$ to be a feasible set, not containing e, hence where the threshold bid $b_e > c_{d(\mathcal{E}\setminus\{e\})}$ then agent e could not have been chosen by VCG in preference to $d(\mathcal{E} \setminus \{e\})$. This gives an upper bound on the threshold payment of $p_e \leq c_{d(\mathcal{E}\setminus\{e\})}$. From Lemma 2.1 we can observe that $c_{d(\mathcal{E}\setminus\{e\})} \leq \text{NTUmin}(\mathbf{c})$, by transitivity we have $p_e \leq \text{NTUmin}(\mathbf{c})$. □

Theorem 2.3. *For all set-system auctions,* $\phi_{\text{NTUmin}(\mathbf{c})}(VCG) \leq n - 1$.

Proof. As we are always assuming a monopoly-free setting, we will have a winning set S such that $|S| \leq n - 1$. We have upper bounds on the payment for each $e \in S$, from Lemma 2.2 this is

$$p_e \leq \text{NTUmin}(\mathbf{c}).$$

So summing up over $e \in S$ gives

$$p_S \leq (n-1)\text{NTUmin}(\mathbf{c}),$$

which completes the proof of the theorem. □

2.3 Frugality of Approximation Algorithms

Let $\mathcal{P}$ be some approximation algorithm, and let $S^{\mathcal{P}}$ be the feasible set returned by $\mathcal{P}$ (which uses the bids as an input parameter). We will assume that $\mathcal{P}$ is monotonic in the bids (that is, given fixed bids of the other agents, no agent can be chosen in the winning set when some smaller bid may result in that agent not being chosen). So if we use this algorithm as a selection rule, and use threshold payments as a payment rule, then it is well-known (e.g. [35]) that we have a resulting truthful mechanism $\mathcal{M}^{\mathcal{P}}$. Let k be the approximation ratio of the algorithm; i.e. some k, such that for all instances of the problem $c_{S^{\mathcal{P}}} \leq k c_S$ holds.

Lemma 2.4. *Let k be the approximation ratio of the algorithm $\mathcal{P}$. Then* $\forall e \in S^{\mathcal{P}}, p_e \leq k\text{NTUmin}(\mathbf{c})$.

Proof. We have defined $d(\mathcal{E} \setminus \{e\})$ to be a (lowest cost) feasible set, not containing e. Assume, for contradiction, that e were to make a threshold bid, $b_e > k\text{NTUmin}(\mathbf{c})$, and the winning set $S^{\mathcal{P}}$ (chosen by $\mathcal{P}$) includes e. Assuming that all other bids are equal to their costs, as the mechanism is truthful, from Lemma 2.1 we can observe that

$b_{d(\mathcal{E}\setminus\{e\})} \leq \text{NTUmin}(\mathbf{c})$. As we have assumed that $b_e \geq k\text{NTUmin}(\mathbf{c})$, and as $e \in S^{\mathcal{P}}$ we have $b_{S^{\mathcal{P}}} > k\text{NTUmin}(\mathbf{c})$ (this holds for all choices of $S^{\mathcal{P}}$ when $e \in S^{\mathcal{P}}$). Hence, by transitivity, we have

$$b_{S^{\mathcal{P}}} > kb_{d(\mathcal{E}\setminus\{e\})}. \tag{2.3}$$

When $b_{S^{\mathcal{P}}} > b_{d(\mathcal{E}\setminus\{e\})}k$, and $d(\mathcal{E}\setminus\{e\})$ is a feasible set, the approximation ratio of $\mathcal{P}$ is at least $\frac{b_{S^{\mathcal{P}}}}{b_{d(\mathcal{E}\setminus\{e\})}}$ and rewriting Inequality 2.3 gives $\frac{b_{S^{\mathcal{P}}}}{b_{d(\mathcal{E}\setminus\{e\})}} > k$, showing that $\mathcal{P}$ does not have an approximation ratio of k, giving a contradiction. Therefore for the threshold bid the inequality $b_e \leq k\text{NTUmin}(\mathbf{c})$ holds, and hence the payment $p_e \leq k\text{NTUmin}(\mathbf{c})$. □

Theorem 2.5. *Let $\mathcal{P}$ be a monotonic approximation algorithm with an approximation ratio of k. Then the resulting mechanism $\mathcal{M}^{\mathcal{P}}$ (with selection rule $\mathcal{P}$ and threshold payments) has $\phi_{\text{NTUmin}(\mathbf{c})}(\mathcal{M}^{\mathcal{P}}) \leq k(n-1)$.*

Proof. As we are always assuming a monopoly-free setting, then we will have a winning set $S^{\mathcal{P}}$ such that $|S| \leq n-1$. We have upper bounds on the payment for each $e \in S$, from Lemma 2.4, this is

$$p_e \leq k\text{NTUmin}(\mathbf{c}).$$

So summing up over $e \in S$ gives

$$p_{S^{\mathcal{P}}} \leq (n-1)k\text{NTUmin}(\mathbf{c})$$

which completes the proof of the theorem. □

2.3.1 Considering the Minimal Winning Sets

In order to avoid some needless technicalities, we wish to, without loss of generality, restrict the winning sets that VCG may choose to be only those that are minimal with regard to set inclusion. We can do this simply by choosing the winning set $S \in \mathcal{F}$ as the lexicographically first set of those that have the lowest cost and also have the smallest cardinality. We now present a proof that any choice of winning set, due to tie-breaking, can be made without changing the payments made.

Proposition 2.6. *For all set-system auctions, having $S \in \mathcal{F}$ and $R \in \mathcal{F}$ as minimum-cost feasible sets then $p_R^{\text{VCG}} = p_S^{\text{VCG}}$.*

Proof. Assume that S is the winning set chosen. For all $e \notin R$, from the definition of a VCG payment, we have $p_e^{VCG} = c_e + c_{T_e} - c_S$ for a minimal cost feasible set T_e such that $e \notin T_e$. From R being a lowest-cost feasible set, we have $c_R \leq c_{T_e}$ and hence $p_e^{VCG} = c_e + c_R - c_S$ giving $p_e^{VCG} = c_e$. Summing for all $e \in S \setminus R$ gives $p_{S\setminus R}^{VCG} = c_{S\setminus R}$.

Similarly, assume that R is the winning set chosen, and re-arrange the labels to give $p_{R\setminus S}^{VCG} = c_{R\setminus S}$.

Additionally, for all $e \in S \cap R$, the threshold bid is not dependent on which of S or R is chosen as the winning set, and hence we have $p_{S\setminus S}^{VCG} + p_{S\cap R}^{VCG} = p_{R\setminus S}^{VCG} + p_{S\cap R}^{VCG}$ giving $p_S^{VCG} = p_R^{VCG}$.

□

Chapter 3

The Single-Commodity Auction

3.1 Overview

In this chapter we examine one of the interesting special-cases of set-system auctions. We will call this a Single-Commodity Auction; in this auction, we have some number of interchangeable items for sale, and a quantity parameter, which we will denote by Q, describing how many of these items the auctioneer requires.

While this auction is relatively simple, it is a reasonable model for the purchase of some number of items, where the exact properties of each may not be an important distinction, provided that they all meet some minimum specification. The purchase of stationery supplies, such as printer paper, seems to be an obvious example for this type of auction, but there are many purchases that can be made of different but interchangeable products.

We will formally define this auction in Section 3.2, and examine a further special case of it (when each agent may only supply at most 2 items) in Section 3.3. There, we will see that the well-known VCG mechanism can overpay, relative to NTUmin, by a factor as large as Q — the number of items required by the buyer.

The measure of overpayment by a (truthful) mechanism is the central theme of this chapter. It is often considered as the additional price that is paid by a mechanism that is truthful, above the price which might be obtained by an optimal, non-truthful, mechanism; sometimes called 'the price of truthfulness'. There was a formal definition for a 'frugality ratio' given in Chapter 1, and it is the primary purpose of this chapter to improve on the frugality ratio of VCG for the Single-Commodity Auction.

In order to give a frugality ratio for a specific mechanism we need to compare the payments of the mechanism with the value NTUmin. In order to do that, we present an algorithm in Section 3.3.3 which computes an NTUmin value, and then use this to describe a characterization of NTUmin with respect to certain bid values. We then use this characterization to provide lower-bounds for NTUmin which can then be used to prove frugality ratios.

In Section 3.4 we discuss a class of alternative, truthful, mechanisms, which we call $\alpha\mathcal{M}$, before showing that one of these (when the parameter $\alpha = \sqrt{Q}$) has a frugality ratio of $2\sqrt{Q}$, compared to Q for VCG. We also see that this is within a factor of 2 of optimal for a particular given class of mechanisms that we describe.

Turning our attention back to the unrestricted single commodity auction, in Section 3.5, we then see a lower bound on the frugality ratio that is possible for a particular class of mechanisms (which includes the $\alpha\mathcal{M}$ mechanism).

We finish this chapter with some observations on the results, and some ideas for how to naturally extend this work.

3.2 Definitions

Here we study the Single-Commodity Auction. In this setting, our auction is for a buyer to purchase some integer quantity, Q, of identical items from a number of sellers. Each seller e will have a specific, publicly known, integer quantity of items to sell that we will call q_e, and we will require that an agent either sells all items or none. The feasible sets, $\mathcal{F}$, are defined from these quantity parameters, as follows;

$$\mathcal{F} = \{T \in 2^{\mathcal{E}} : \left(\sum_{e \in T} q_e\right) \geq Q\}. \tag{3.1}$$

We will refer to the sellers as 'agents', and we will implement our auction as a set system auction, which was described in Section 1.2.1.

As a reminder, each agent e will submit a sealed bid b_e to the auction process. The auctioneer will then, by means of a mechanism $\mathcal{M}$, use its selection rule to choose one of the feasible sets, which we will call $S^{\mathcal{M}} \in \mathcal{F}$ to be the 'winners'. The mechanism will then pay each agent $e \in S^{\mathcal{M}}$ a value $p_e^{\mathcal{M}}$, calculated using some payment rule.

3.3 The $\{1, 2\}$ Single-Commodity Auction

In the first part of this chapter, we will impose an additional restriction — that each agent e will only have for sale at most 2 items. We call this the {1,2} Single-Commodity Auction.

3.3.1 Auction Definition

The $\{1, 2\}$ Single-Commodity Auction is a single commodity auction, as defined in Section 3.2. The additional restriction, that the $\{1, 2\}$ Single-Commodity Auction must meet is that all agents have a quantity value of at most 2,

$$\forall e \in \mathcal{E}, q_e \in \{1, 2\}.$$

While we could simply use VCG to run this auction (VCG chooses the lowest-cost solution and pays each winning agent a threshold value), we will see later that this performs poorly in terms of frugality. In an attempt to improve frugality, we will now look at a class of (truthful) mechanisms that choose a winning set a little more intelligently.

3.3.2 The $\alpha\mathcal{M}$ Mechanism

Here we will be analysing a class of mechanisms, $\alpha\mathcal{M}$, each of which is uniquely defined by its 'scaling' value $\alpha \geq 1$; a definition for this mechanism follows.

$\alpha\mathcal{M}$ will calculate 'virtual' bids v_e for each agent e by using a scaling factor as follows

$$v_e = \begin{cases} \alpha b_e, & \text{if } q_e = 1 \\ b_e, & \text{otherwise.} \end{cases}$$

In order to simplify some of the later notation, we define the aggregate to be the sum of virtual bids over any set of agents, V,

$$v_V = \sum_{e \in V} v_e.$$

Let $S^\alpha \in \operatorname{argmin}_{T \in \mathcal{F}} v_T$ be the winning set. For concreteness, let S^α be the lexicographically first of the feasible sets that have the lowest sum of virtual bids. It is easy to observe that this selection will be monotonic in the bids, which is known to be necessary for a mechanism to be truthful (see, e.g., [35]).

The payment rule is simple — each agent $e \in S^\alpha$ will be paid its threshold bid. This is the supremum of the amounts that the agent can bid and still be selected in the winning set, given the fixed bids of the other agents. It is also well-known (see, e.g. [34, 17]) that this threshold payment is required in order for the mechanism to be truthful, and that a mechanism with a monotonic selection rule and threshold payments is a truthful mechanism.

Let $m = |S^\alpha|$ and let the payment to agent e by the $\alpha\mathcal{M}$ mechanism be denoted by p_e^α, resulting in the payment vector $\mathbf{p}^\alpha = (p_1^\alpha, \ldots, p_m^\alpha)$. Observe that when $\alpha = 1$ the $\alpha\mathcal{M}$ mechanism is exactly equivalent to VCG. The VCG mechanism chooses the feasible set which has the lowest sum of bids (which are equal to the costs, as VCG is known to be truthful, see, e.g.,[35]). Each of these 'winning' agents is paid its threshold value — the value of the largest bid the agent could have submitted but still remain in the winning set, given all other bids being unchanged. Interestingly, we can see in Example 3.1 below that the frugality $\phi_{\mathrm{NTUmin}}(\mathrm{VCG}) \geq Q$ even for this restricted setting, so we have a strong motivation to find a mechanism with lower frugality.

Example 3.1. *In this example we will see that VCG has poor frugality; we have a single commodity auction for quantity Q items and observe that the number of agents $n = Q+1$. For each agent $e \in \mathcal{E}$ the quantity q_e and cost c_e are given in Table 3.1. Here we can see* *that, when $S = \{1, \ldots, n-1\}$, the bid vector* $\mathbf{b}^{\min}$ *meets conditions (1),(2) and (3) for a* NTUmin *vector. To verify this, for all $e \in S \setminus \{A_1\}$, then $T_e = S \setminus \{A_1\} \cup \{A_n\}$. This satisfies condition (3) in Definition 1.2 as $b_e + b_1 = c_n$ (and any such T_e set satisfies condition (3) for A_1). Conditions (1) and (2) are easily verifiable, showing that* NTUmin ≤ 1. *The payment vector* $\mathbf{p}^{\alpha}$ *corresponds to the payments made by the $\alpha\mathcal{M}$ mechanism, hence $p^{\alpha} \geq Q$ and* NTUmin ≤ 1 *giving* $\frac{p^{\alpha}}{\text{NTUmin}} \geq Q$.

	Agent	q_e	c_e	$b_e^{\min}$	p_e^{VCG}
	A_1	1	0	1	1
	A_2	1	0	0	1
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	A_{n-1}	1	0	0	1
	A_n	2	1		
Total				1	$n-1$

TABLE 3.1: Instance I with VCG ($\alpha = 1$)

Adding some additional notation, let $\mathcal{E}_1 \subseteq \mathcal{E}$ be the subset of all agents having quantity 1, and let $\mathcal{E}_2 = \mathcal{E} \setminus \mathcal{E}_1$ be the subset of agents having quantity 2. Observe that this implies $q_{\mathcal{E}_1} = |\mathcal{E}_1|$ and $q_{\mathcal{E}_2} = 2|\mathcal{E}_2|$.

We only consider the minimal winning sets, with respect to set inclusion. Proposition 2.6 shows us that this is without loss of generality.

3.3.3 Computing NTUmin for the {1,2} Single-Commodity Auction

Let $\mathbf{c}$ be the cost vector of a fixed instance of the {1,2} Single-Commodity Auction. We will now see a method of computing the NTUmin bid value for all $e \in S$.

We will firstly select a winning set S, as a feasible set with the lowest total cost. We can do this by the use of a *knapsack* algorithm. A standard knapsack problem is, given a capacity C and a set of n items, with each item $i \in \{1, \ldots, n\}$ having value v_i and weight w_i, to find the set of items with the largest sum of values such that the sum of weights does not exceed the capacity. Assuming integer weights and capacity, it is well-known to be solvable in $O(nC)$ time [15]. An integer commodity auction can be translated to a knapsack problem as follows. Let $C = Q_{\mathcal{E}} - Q$, $w_i = q_e$ and $v_i = c_e$ and hence the knapsack algorithm will return a maximum value set T, having weight at most $Q_{\mathcal{E}} - Q$, leaving $S = \mathcal{E} \setminus T$ as a lowest-cost set with quantity at least Q.

We will then present Algorithm 1 that will, for the agents in the winning set S, calculate a bid vector $\mathbf{b} = (b_1, \ldots, b_m)$. We will analyse the properties of any valid NTUmin bid vector, and we will see a proof that the bid vector $\mathbf{b}$ output by the algorithm qualifies as a NTUmin bid vector. We will then see in Theorem 3.12 how the NTUmin value may be given by a number of expressions, based on the composition of the cost vector $\mathbf{c}$.

Preliminaries

Fix S as a single winning solution. (Remember that, if there are ties, the NTUmin value is computed from any of these choices of S). For ease of notation, let $S_1 = S \cap \mathcal{E}_1$ and let $S_2 = S \cap \mathcal{E}_1$; similarly when $S^{\mathcal{M}}$ is a set chosen by mechanism $\mathcal{M}$, let $S_1^{\mathcal{M}} = S^{\mathcal{M}} \cap \mathcal{E}_1$ and let $S_2^{\mathcal{M}} = S^{\mathcal{M}} \cap \mathcal{E}_2$.

In order to examine NTUmin we will define some 'replacement' sets, as follows.

Let $R_x \subseteq (\mathcal{E} \backslash S)$ be a subset of non-elements of S when $q_{R_x} \geq x$ and c_{R_x} is minimized. For concreteness, if there is more than one suitable choice, let R_x be the lexicographically first. This will be used in this general form in Proposition 3.1, but more frequently as one of the following specific forms.

Let $R_1 \subseteq (\mathcal{E} \setminus S)$ be the subset from outside S when $q_{R_1} \geq 1$ and c_{R_1} is minimized. Note that R_1 is only empty when S is the only feasible set, and hence each agent has a monopoly. As we are only interested in monopoly-free set systems, we can therefore assume that R_1 is non-empty.

Let $R_2 \subseteq (\mathcal{E} \setminus S)$ be the subset from outside S when $q_{R_2} \geq 2$ and c_{R_2} is minimized. Observe that R_2 may be empty, in a monopoly-free set-system, only if $Q = q_{\mathcal{E}} - 1$. When the lowest-cost agent outside S has quantity 2, we may have $R_1 = R_2$.

Let $R_2^* \subseteq (\mathcal{E}_2 \setminus S)$ be the set containing the lowest-cost agent with quantity 2 from non-members of S. Note that this may be empty even when there is no monopoly.

Describing the NTUmin bids.

Recall Definition 1.2, the definition of NTUmin given in Chapter 1. In this section, we give an algorithm to compute a bid vector $\mathbf{b}^{\min}$ that meets the definition of NTUmin for all instances of the {1,2} Single-Commodity Auction. We then use the result of this algorithm to describe the NTUmin value in terms of the agents' bids.

Firstly, for each agent e, we will see an upper bound on the bid $b_e^{\min}$ of that agent e. This upper bound is simply the lowest cost of some subset $R_{q_e} \subseteq \mathcal{E} \setminus S$ of non-members of S that could 'replace' the agent e and make a feasible set.

We then address the following technical difficulty; there may be many sets for T_e that meet the definition for condition (3) in Definition 1.2, and some of these T_e sets may be unnecessarily different to S. We will see in Proposition 3.2 a result that implies, when calculating NTUmin values, we only need to be concerned with the 'minimally different' T_e sets — a precise definition of this is given there. Intuitively, this occurs when T_e has chosen some agents, due to tie-breaking, but in choosing S different agents were chosen in the tie-breaking.

We will then use these results, in Lemma 3.4 and Lemma 3.5, to define the payments given to all agents with quantity 2. Lemma 3.5 deals with a special case, when Q is odd and S is made up of agents each with quantity 2, and hence $q_S > Q$. Following that, there are several lemmas (3.6, 3.8, 3.9, and 3.10) that give the bids of agents with quantity 1 under different conditions.

Our first proposition provides a straightforward upper bound on the NTUmin bids for sets of agents; any subset of winning agents may not bid more than some possible replacement from outside S. Recall from Chapter 1 the definition $b_V^{\min} = \sum_{e \in V} b_e^{\min}$.

Proposition 3.1. *For any subset $V \subseteq S$ having $q_{\mathcal{E} \setminus S} \geq q_V$, the inequality $b_V^{\min} \leq c_{R_{q_V}}$ holds.*

Proof. Our assumption that $q_{\mathcal{E} \setminus S} \geq q_V$ provides that there is a non-empty assignment for R_{q_V}. Let $W = (S \setminus V) \cup R_{q_V}$. We can observe that W is a feasible set as follows; from the definition of R, R_{q_V} must be large enough to replace quantity q_V and still make a feasible set. We can write this as

$$q_{R_{q_V}} \geq q_V \tag{3.2}$$

and we can expand q_W to get

$$q_W = q_S - q_V + q_{R_{q_V}}$$

and substituting the inequality of 3.2 we can rewrite this as $q_W \geq q_S$, showing that W is a feasible set.

Therefore, we must have $b_{S \setminus W}^{\min} \leq c_{W \setminus S}$ or else W violates condition (2) in Definition 1.2. By substituting from the definition of W, this can be written as $b_V^{\min} \leq c_{R_{q_V}}$. Hence any bid vector that does not satisfy this inequality must violate the constraint in condition (2) and hence cannot be a valid NTUmin bid vector.

□

We will now see that when $q_{T_e \setminus S}$ is larger than necessary, for a T_e set satisfying condition (3) in Definition 1.2, then there is always some other set T'_e with a smaller $q_{T'_e \setminus S}$, and that this 'less different' set also satisfies condition (3). We will use this to show that we can restrict our analysis to the 'less different' sets, and hence finally the 'minimally different' sets, which are those that have the largest intersection with the winning set S. Informally, this is quite obvious — when we may have some T_e set given which contains some agents that are outside S, but could have also chosen them from inside S instead (in the event of a tie), we don't need to consider that T_e set, but would prefer to analyse the one that chose agents inside S, when possible.

Proposition 3.2. *For every set T_e that satisfies condition (3) in Definition 1.2 and has $q_{T_e \setminus S} \geq 3$, there exists another feasible set T'_e when the equation $q_{T'_e \setminus S} = q_{T_e \setminus S} - 2$ holds and T'_e satisfies condition (3) in Definition 1.2.*

Proof. Observe that any set of quantity at least 2 must contain a subset of quantity 2. Therefore there exists a subset $Y \subset (T_e \setminus S)$ such that $q_Y = 2$. As $q_{T_e \setminus S} \geq 3$, we have

$q_{S\setminus T_e} \geq 2$ (or else $q_S < q_{T_e} - 1$ implying S is not a feasible set). Hence there is some subset $Y' \subset (S \setminus T_e)$ such that $q_{Y'} = 2$.

Let $T_Y = (S \setminus Y') \cup Y$ be a feasible set (it has quantity q_S). As T_Y must satisfy condition (2) in Definition 1.2, we have

$$b^{\min}_{S\setminus T_Y} \leq c_{T_Y\setminus S}$$

which can be re-expressed as

$$b^{\min}_{S\setminus((S\setminus Y')\cup Y)} \leq c_{((S\setminus Y')\cup Y)\setminus S}$$

and (as $Y' \subset S$ and $Y \subset (T_Y \setminus S)$)

$$b^{\min}_{Y'} \leq c_Y. \tag{3.3}$$

Let $T'_e = (T_e \setminus Y) \cup Y'$ be a feasible set (it has quantity q_{T_e}), which must also satisfy condition (2) in Definition 1.2, we can rewrite this as

$$b^{\min}_{(S\setminus T_e)\setminus Y'} \leq c_{(T_e\setminus S)\setminus Y} \tag{3.4}$$

As we have assumed that T_e satisfies condition (3) in Definition 1.2, we can write this as

$$b^{\min}_{S\setminus T_e} = c_{T_e\setminus S} \tag{3.5}$$

and we can trivially observe that $S\setminus T_e = ((S\setminus T_e)\setminus Y')\cup Y'$ and $T_e\setminus S = ((T_Y\setminus S)\setminus Y)\cup Y$, which shows that the sets in inequalities 3.3 and 3.4 make partitions of the sets in Equation 3.5. Therefore Inequality 3.3 and Equation 3.4 can be strengthened to give

$$b^{\min}_{Y'} = c_Y \tag{3.6}$$

and

$$b^{\min}_{(S\setminus T_e)\setminus Y'} = c_{(T_e\setminus S)\setminus Y} \tag{3.7}$$

Equation 3.7 can be rewritten to give

$$b^{\min}_{S\setminus T'_e} = c_{T'_e\setminus S}$$

showing that set T'_e satisfies (3), and we can see from its definition that $q_{T'_e\setminus S} = q_{T_e\setminus S} - 2$.
□

A similar proposition can be proven for some cases when $T_e \setminus S$ contains an agent with quantity 1. Recall the definition that $S_1 = S \cap \mathcal{E}_1$.

Proposition 3.3. *Suppose there is a feasible set T_e that satisfies condition (3) in Definition 1.2, $q_S = Q$, $q_{T_e\setminus S} \geq 3$, and $q_{T_e\setminus S}$ is odd. Then there exists another feasible set T'_e when the equation $q_{T'_e\setminus S} = q_{T_e\setminus S} - 1$ holds and T'_e satisfies condition (3)in Definition 1.2.*

Proof. Observe that there exists a subset $Y \subset (T_e \setminus S)$ such that $q_Y = 1$ (as $q_{T_e \setminus S}$ is odd).

As T_e contains an agent with quantity 1, hence $q_{T_e} = Q = q_S$. This implies $q_{T_e \setminus S} = q_{S \setminus T_e}$ and hence $S \setminus T_e$ also contains an agent with quantity 1. Therefore there is also some other subset $Y' \subset (S \setminus T_e)$ such that $q_{Y'} = 1$.

Let $T_Y = S \setminus Y' \cup Y$ be a feasible set (it has quantity of q_S). As T_Y must satisfy condition (2) in Definition 1.2, we have

$$b^{\min}_{S \setminus T_Y} \leq c_{T_Y \setminus S}$$

which can be simplified to

$$b^{\min}_{Y'} \leq c_Y. \tag{3.8}$$

Let $T_e' = (T_e \setminus Y) \cup Y'$ be a feasible set (it has quantity of q_{T_e}), which must also satisfy condition (2) in Definition 1.2, we can rewrite this as

$$b^{\min}_{(S \setminus T_e) \setminus Y'} \leq c_{(T_e \setminus S) \setminus Y}. \tag{3.9}$$

As we have assumed that T_e satisfies condition (3), we can write this as

$$b^{\min}_{S \setminus T_e} = c_{T_e \setminus S} \tag{3.10}$$

we have seen that the sets in inequalities 3.8 and 3.9 make partitions of the sets in Equation 3.10. Therefore Inequality 3.8 and Inequality 3.9 can be strengthened to give

$$b^{\min}_{Y'} = c_Y \tag{3.11}$$

and

$$b^{\min}_{(S \setminus T_e) \setminus Y'} = c_{(T_e \setminus S) \setminus Y}. \tag{3.12}$$

Equation 3.12 can be rewritten to give

$$b^{\min}_{S \setminus T_e'} = c_{T_e' \setminus S}$$

showing that set T_e' satisfies (3).

□

We will now proceed to examine the NTUmin bid values directly. Recall that Q is the quantity required by the buyer, and observe that when $q_S \neq Q$ there are no agents in S with quantity 1, and Q is odd. (We have previously discounted solutions when superfluous agents with cost 0 have been included). We will now see a characterization of the NTUmin bid values for the agents having quantity 2, as separate cases for $q_S = Q$ and $q_S \neq Q$.

Lemma 3.4. *Given a $\{1, 2\}$ single-commodity auction having $q_S = Q$, for every agent $e \in S_2$ the equation $b^{\min}_e = c_{R_2}$ holds.*

Proof. Proposition 3.1 gives us $b_e^{\min} \leq c_{R_2}$. For the other direction, we will see this as two cases. Firstly, for some T_e satisfying (3), when $q_{T_e \setminus S} = 2$ then (3) gives us $b_{S \setminus T_e}^{\min} = c_{T_e \setminus S}$, and by the minimality of R_2, we have $b_{S \setminus T_e}^{\min} \geq c_{R_2}$, which can be simplified to $b_e^{\min} \geq c_{R_2}$.

Secondly, when $q_{T_e \setminus S} > 2$ either Proposition 3.3 or Proposition 3.2 may be applied (possibly repeatedly) to give some set T'_e such that $q_{T'_e \setminus S} = 2$ and T'_e satisfies condition (3), giving $b_{S \setminus T_e}^{\min} \geq c_{R_2}$ and hence $b_e^{\min} \geq c_{R_2}$.

This has shown that $b_e^{\min} \leq c_{R_2}$ and $b_e^{\min} \geq c_{R_2}$ hold, therefore we have $b_e^{\min} = c_{R_2}$. □

Lemma 3.5. *Given a* $\{1, 2\}$ *single-commodity auction having* $q_S = Q + 1$*, for every agent* $e \in S$ *the equation* $b_e^{\min} = c_{R_1}$ *holds.*

Proof. Observe that, as the largest quantity of any agent is 2, then $q_S < Q + 2$. This implies that there are no agents in S with quantity 1 (as they would not be needed, and we assume that we do not include superfluous agents).

Let $T = S \setminus \{e\} \cup R_1$; observe that $T \in \mathcal{F}$ and so Proposition 3.1 gives us $b_e^{\min} \leq c_{R_1}$. Assume some T_e set that satisfies condition (3). As $q_S = Q + 1$, then $q_{T_e} \geq q_S - 1$ must hold. Hence we must have $q_{T_e \setminus S} \geq q_{S \setminus T_e} - 1$. We will now examine this as three cases.

Case 1: $q_{T_e \setminus S} = 1$

If $q_{T_e \setminus S} = 1$ then $q_{S \setminus T_e} \leq 2$, hence $b_e^{\min} = c_{T_e \setminus S}$ which gives $b_e^{\min} \geq c_{R_1}$ by the minimality of R_1.

Case 2: $q_{T_e \setminus S} = 2$

If $q_{T_e \setminus S} = 2$ then $q_{S \setminus T_e} \leq 3$, and as there are no agents in S with quantity 1, we have $b_e^{\min} = c_{T_e \setminus S}$ and hence $b_e^{\min} \geq c_{R_1}$.

Case 3: $q_{T_e \setminus S} \geq 3$

If $q_{T_e \setminus S} \geq 3$ then we can use Proposition 3.2 (possibly repeatedly) to show that there is some T'_e such that $q_{T'_e \setminus S} \leq 2$ and case 1 or case 2 above must apply.

Now we have seen that both $b_e^{\min} \leq c_{R_1}$ and $b_e^{\min} \geq c_{R_1}$ are proven, showing that $b_e^{\min} = c_{R_1}$ as claimed. □

Now that we have seen the bid values described for agents with quantity 2, so we will move on to examine the bid values for the agents with quantity 1. For $|S_1| \geq 2$ we will do this as three separate cases, which depend on the relationship between certain cost values of the instance. $|S_1| = 1$ will be treated as a (simple) special case.

Lemma 3.6. *Given any* NTUmin *bid vector* $\mathrm{b}^{\min}$ *having* $c_{R_1} < c_{R_2}/2$*, for all* $e \in S_1$*, the equation* $b_e^{\min} = c_{R_1}$ *holds.*

Proof. When $S_1 \neq \emptyset$ we have $q_S = Q$, or else there is some superfluous agent $e \in S$ that would not have been included. From condition (3) there is some T_e such that $b^{\min}_{S \setminus T_e} = c_{T_e \setminus S}$, which we will examine as three cases.

Case 1: $q_{T_e \setminus S} = 1$

As $q_{T_e} = Q$, by T_e including some agent with quantity 1, then we have $q_{S \setminus T_e} \leq 1$ (if not, this would imply that $q_{T_e} < q_S = Q$). This gives $S \setminus T_e = \{e\}$, and hence $b^{\min}_e = c_{T_e \setminus S}$, and by definition of R_1, that $b^{\min}_e \geq c_{R_1}$.

Case 2: $q_{T_e \setminus S} = 2$

We will see that there are actually no cases, satisfying the assumptions, when this holds. For contradiction, assume that $q_{T_e \setminus S} = 2$, therefore $q_{S \setminus T_e} = 2$ (if not, this would imply that $q_{T_e} < q_S$) and there exists some j such that $b^{\min}_e + b^{\min}_j = c_{T_e \setminus S}$. From Proposition 3.1, $b^{\min}_e \leq c_{R_1}$, so we can rewrite to give $b^{\min}_j \geq c_{T_e \setminus S} - c_{R_1}$. As, by definition, $c_{R_2} \leq c_{T_e \setminus S}$ this can be substituted to give $b^{\min}_j \geq c_{R_2} - c_{R_1}$. As the assumption for this lemma states that $c_{R_1} < c_{R_2}/2$, we have $b^{\min}_j > c_{R_2} - c_{R_2}/2$, hence $b^{\min}_j > c_{R_2}/2$, and from the lemma again $b^{\min}_j > c_{R_1}$ contradicting Proposition 3.1.

Case 3: $q_{T_e \setminus S} \geq 3$

When $q_{T_e \setminus S} > 2$, from Proposition 3.2, there exists some other feasible set T'_e such that T'_e satisfies condition (3). Applying Proposition 3.2 repeatedly will give some T'_e set such that $q_{T'_e \setminus S} \leq 2$, and hence satisfies the definition for either case 1 or case 2 above (although we have seen that only case 1 is possible).

This has shown that, when $c_{R_1} < c_{R_2}/2$ holds, for every agent $e \in S_1$ then every possible set T_e satisfying (3) gives $b^{\min}_e \geq c_{R_1}$, and from Proposition 3.1 we have $b^{\min}_e \leq c_{R_1}$, with both together giving $b^{\min}_e = c_{R_1}$ as claimed.

□

The other two cases will both make use of the following proposition, which shows that there is a limit on the sum of the bids of two agents that each have quantity 1, and that this limit is always reached provided that $c_{R_1} \geq c_{R_2}/2$.

Proposition 3.7. *Let* $\mathbf{b}^{\min}$ *be a* NTUmin *bid vector and suppose the inequalities* $c_{R_1} \geq c_{R_2}/2$ *and* $|S_1| \geq 2$ *hold. Then, for every agent* $e \in S_1$, $b^{\min}_e = c_{R_2} - \max_{j \in (S_1 \setminus \{e\})} b^{\min}_j$.

Proof. Firstly, assume for contradiction that $b^{\min}_e > c_{R_2} - \max_{j \in (S_1) \setminus \{e\}} b^{\min}_j$. This gives $b^{\min}_e + b^{\min}_j > c_{R_2}$, which contradicts Proposition 3.1. For the other direction, assume for contradiction that $b^{\min}_e < c_{R_2} - \max_{j \in (S_1) \setminus \{e\}} b^{\min}_j$.

Using a similar method as in Lemma 3.6, we will examine the possible T_e sets satisfying (3) for every agent $e \in S_1$.

Case 1: $q_{T_e \setminus S} = 1$

Assume, for contradiction, that there exists some other agent, j, such that $q_{T_j \setminus S} = 1$. This would give $b_e^{\min} \geq c_{T_e \setminus S}$ and $b_j^{\min} \geq c_{T_j \setminus S}$. By the minimality of R_1, we would have $b_e^{\min} + b_j^{\min} \geq 2c_{R_1}$. For this proposition, we assume that $c_{R_1} \geq c_{R_2}/2$, which would give $b_e^{\min} + b_j^{\min} > c_{R_2}$ giving a contradiction with Proposition 3.1.

This shows that case 1 may be true, but for at most one agent.

Case 2: $q_{T_e \setminus S} = 2$

This gives $b_e^{\min} + b_j^{\min} = c_{T_e \setminus S}$, and hence $b_e^{\min} + b_j^{\min} \geq c_{R_2}$ from the minimality of R_2.

Case 3: $q_{T_e \setminus S} \geq 3$

When $q_{T_e \setminus S} > 2$, from Proposition 3.2, there exists some other feasible set T'_e such that T'_e satisfies condition (3). Applying Proposition 3.2 repeatedly will give some T'_e set such that $q_{T_e \setminus S} \leq 2$, and hence either case 1 or case 2 above applies.

As case 1 can hold for only one agent, the application of case 2 to all other agents in $\mathcal{E}_1$ is sufficient to apply to every possible pair of agents, and thus completes the proof. □

Finally, we need to consider the cost values of the two agents in S_1 with largest cost. To that end, there are some definitions that will be used; Let ℓ_1 be an agent in S_1 with highest cost, more formally

$$\ell_1 \in \operatorname*{argmax}_{e \in S_1} c_e$$

and let ℓ_2 be an agent in S_1 with second highest cost;

$$\ell_2 \in \operatorname*{argmax}_{e \in (S_1 \setminus \{\ell_1\})} c_e.$$

Lemma 3.8. *Suppose that* $|S_1| \geq 2$ *and* $c_{R_2}/2 < c_{R_1} \leq c_{R_2} - c_{\ell_2}$. *Then there exists a* NTUmin *bid vector* $\mathbf{b}^{\min}$ *such that there is exactly one agent* $e \in S_1$ *having* $b_e^{\min} = c_{R_1}$ *and every other agent* $j \in (S_1) \setminus \{e\}$ *has* $b_j^{\min} = c_{R_2} - c_{R_1}$.

Proof. Recall that we can assume $q_S = Q$, and Proposition 3.7 states that for every agent $e \in S_1$, the following holds;

$$b_e^{\min} = c_{R_2} - \max_{j \in (S_1) \setminus \{e\}} b_j.$$

For any bid vector, $\mathbf{b}$, let $\ell \in \operatorname{argmax}_{e \in S_1} b_e$ be an agent with quantity 1 that has the largest bid, and hence $b_\ell = \max_{e \in S_1} b_e$ is the largest bid of any agent with quantity 1. Observe that Proposition 3.7 implies that $\forall e \in (S_1 \setminus \ell), b_e = c_{R_2} - b_\ell$.

We now claim that, in an NTUmin bid vector $\mathbf{b}^{\min}$, this largest bid $b_\ell^{\min}$ must be equal to c_{R_1}. Proposition 3.1 tells us that $b_\ell^{\min} \leq c_{R_1}$. Now assume for contradiction that when $|S_1| > 2$ there exists a bid vector $\mathbf{b}$ such that $b_S \leq b_S^{\min}$ and that $b_\ell < b_\ell^{\min}$.

Let $\epsilon = b_\ell^{\min} - b_\ell$ (giving $b_\ell = b_\ell^{\min} - \epsilon$) and that, due to Proposition 3.7, this implies that

$$\forall e \in (S_1 \setminus \{\ell\}), b_e = b_e^{\min} + \epsilon.$$

Summing up over S_1 gives

$$b_{S_1} = b_{S_1}^{\min} + (|S_1| - 1)\epsilon.$$

As we have assumed $|S_1| > 2$, this gives $b_{S_1} > b_{S_1}^{\min}$. Observe that any T_e set that has agents of both quantity 1 and 2 in $T_e \setminus S$ must have $q_{T_e \setminus S} \geq 3$, and hence Lemma 3.2 shows that there is some other set T_e' (with $q_{T_e' \setminus S} < 2$) which determines the bid values of $S \setminus T_e'$. This shows that the bids of agents with quantity 1 are not dependent on the bids of agents with quantity 2 and hence $b_{S_1} > b_{S_1}^{\min}$ contradicts $b_S \leq b_S^{\min}$ (as $b_{S_2}^{\min} \leq b_{S_2}$).

As we have seen that $b_\ell = c_{R_1}$, then Proposition 3.7 shows that every other agent $j \in (S_1 \setminus \{\ell\})$ must bid $b_j = c_{R_2} - c_{R_1}$. As this concludes the proof when $|S_1| > 2$, we finish with $|S_1| = 2$. Due to Proposition 3.7, we must have $b_{S_1} = c_{R_2}$, which may be achieved with two bids of $(c_{R_1}, c_{R_2} - c_{R_1})$. (Although there are other bid vectors that would satisfy NTUmin.)

□

Lemma 3.9. *Suppose* $|S_1| \geq 2$ *and* $c_{R_1} > c_{R_2} - c_{\ell_2}$. *Then there exists a* NTUmin *bid vector* $\mathbf{b}^{\min}$ *such that there is exactly one agent* $e \in S_1$ *having* $b_e^{\min} = c_{R_2} - c_{\ell_2}$ *and every other agent* $j \in (S_1) \setminus \{e\}$ *has* $b_j^{\min} = c_{\ell_2}$.

Proof. No agent $e \in S_1$ may bid $b_e^{\min} > c_{R_2} - c_{\ell_2}$ or else there exists some $j \in \{\ell_1, \ell_2\}$ such that $b_e^{\min} + b_j^{\min} > c_{R_2}$ and Proposition 3.1 would show that this contradicts $\mathbf{b}^{\min}$ being a valid NTUmin bid vector.

Now we will see that, when $|S_1| > 2$ there is some agent $e \in S_1$ that must bid $b_e^{\min} = c_{R_2} - c_{\ell_2}$.

As in Lemma 3.8, let $\ell = \operatorname{argmax}_{e \in S_1} b_e$ be the agent with quantity 1 that has the largest bid, and hence $b_\ell = \max_{e \in S_1} b_e$ is the largest bid of any agent with quantity 1. Observe that Proposition 3.7 implies that $\forall e \in (S_1 \setminus \{\ell\}), b_e = c_{R_2} - b_\ell$.

We now claim that, in an NTUmin bid vector $\mathbf{b}^{\min}$, this largest bid $b_\ell^{\min}$ must be equal to $c_{R_2} - c_{\ell_2}$. Assume for contradiction that when $|S_1| > 2$ there exists a bid vector $\mathbf{b}$ such that $b_S \leq b_S^{\min}$ and that $b_\ell < b_\ell^{\min}$. Let $\epsilon = b_\ell^{\min} - b_\ell$ (giving $b_\ell = b_\ell^{\min} - \epsilon$) and that, due to Proposition 3.7, this implies that

$$\forall e \in S_1 \setminus \{\ell\}, b_e = b_e^{\min} + \epsilon$$

Summing up over S_1 gives

$$b_{S_1} = b_{S_1}^{\min} + (|S_1| - 1)\epsilon$$

As we have assumed $|S_1| > 2$, this gives $b_{S_1} > b_{S_1}^{\min}$, giving a contradiction.

Recall that the bids of agents with quantity 1 are not dependent on the bids of agents with quantity 2 and therefore $b_{S_1} > b_{S_1}^{\min}$ contradicts $b_S \leq b_S^{\min}$. As we have seen that

$b_\ell = c_{R_2} - c_{\ell_2}$, then Proposition 3.7 shows that every other agent $j \in (S_1 \setminus \{\ell\})$ must bid $b_j = c_{\ell_2}$.

Finally, when $|S_1| = 2$, then due to Proposition 3.7, we must have $b_{S_1} = c_{R_2}$, which may be achieved with two bids of $(c_{\ell_2}, c_{R_2} - c_{\ell_2})$.

□

We now finish the bids of agents with quantity 1, by looking at the remaining case when there is only a single agent with quantity 1 in the winning set.

Lemma 3.10. *For all $e \in S_1$, the equation $b_e^{\min} = c_{R_1}$ holds when $|S_1| = 1$.*

Proof. Again, using a similar method to Lemma 3.6, we will examine the possible T_e sets satisfying (3) for the agent $e \in S_1$.

Case 1: $q_{T_e \setminus S} = 1$

This gives $b^{\min}_{S \setminus T_e} = c_{T_e \setminus S}$. As S_1 is non-empty, we have $q_S = Q$, and hence $q_{S \setminus T_e} = q_{T_e \setminus S} = 1$, and hence $b_e^{\min} = c_{T_e \setminus S}$. From the minimality of R_1, we have $b_e^{\min} \geq c_{R_1}$. The other direction, $b_e^{\min} \leq c_{R_1}$ is shown by Proposition 3.1, giving $b_e^{\min} = c_{R_1}$.

Case 2: $q_{T_e \setminus S} = 2$

Observe that there is no T_e set where $q_{S \setminus T_e} = 2$ (as there is no other agent with quantity 1).

Case 3: $q_{T_e \setminus S} \geq 3$

When $q_{T_e \setminus S} > 2$ then, from Proposition 3.2, there exists some other feasible set T'_e with $q_{T'_e \setminus S} < q_{T_e \setminus S}$ such that T'_e satisfies condition (3). Applying Proposition 3.2 repeatedly will give some T'_e set such that $q_{T_e \setminus S} \leq 2$, and hence either case 1 or case 2 above can be applied.

□

Computing the NTUmin bid values

We propose a simple algorithm, Algorithm 1, which computes an NTUmin bid vector. We use the results of the lemmas in the previous section to verify that Algorithm 1 correctly calculates a bid vector $\mathbf{b}^{\min}$ that satisfies the definition of NTUmin. Recall that S is a lowest cost solution.

Lemma 3.11. *Algorithm 1 computes a* NTUmin *bid vector* $\mathbf{b}^{\min}$ *for any cost vector* $\mathbf{c}$ *of a {1,2} Single-Commodity Auction.*

Proof. If there is at least one agent with quantity 1, then exactly one element ℓ is assigned $b_\ell^{\min} = \min(c_{R_1}, c_{R_2} - c_{\ell_2})$; all other elements with quantity 1 are assigned $\min(c_{R_1}, c_{R_2} - b_\ell^{\min})$. This requirement was proven in Lemmas 3.6, 3.8, 3.9, and 3.10.

All other agents with quantity 2 are assigned c_{R_2}, which was proven to be correct in Lemma 3.4. Hence, as each agent is assigned a value that is consistent with the values

Algorithm 1: Algorithm to calculate NTUmin bids

$R_1 \leftarrow \text{argmin}_R\{c_R : R \subseteq \mathcal{E} \setminus S, q_R = 1\}$;
$R_2 \leftarrow \text{argmin}_R\{c_R : R \subseteq \mathcal{E} \setminus S, q_R = 2\}$;
$\ell_1 \leftarrow \text{argmax}_e\{c_e : e \in S_1\}$;
$\ell_2 \leftarrow \text{argmax}_e\{c_e : e \in (S_1 \setminus \{\ell_1\})\}$;
if $|S_1| > 0$ **then**
    $b_{\ell_1}^{\min} \leftarrow \min(c_{R_1}, c_{R_2} - c_{\ell_2})$;
    for each $i \in (S_1) \setminus \{\ell_1\}$ **do**
        $b_i^{\min} \leftarrow \min(c_{R_1}, c_{R_2} - b_{\ell_1}^{\min})$;
for each $j \in S_2$ **do**
    if $q_S = Q$ **then**
        $b_j^{\min} \leftarrow c_{R_2}$;
    else
        $b_j^{\min} \leftarrow c_{R_1}$;
Return $\mathbf{b}^{\min}$;

required for a NTUmin bid vector, then Algorithm 1 correctly calculates a NTUmin bid vector $\mathbf{b}^{\min}$.

□

Theorem 3.12. *For a* $\{1, 2\}$ *single commodity auction,* $\text{NTUmin}(\mathbf{c})$ *is given by one of the following expressions.*

Case 1: If $|S_1| = 0$ *and* $q_S = Q$ *then*

$$\text{NTUmin}(\mathbf{c}) = |S|c_{R_2}. \tag{3.13}$$

Case 2: If $|S_1| = 0$ *and* $q_S > Q$ *then*

$$\text{NTUmin}(\mathbf{c}) = |S|c_{R_1}. \tag{3.14}$$

Case 3: If $|S_1| = 1$ *then*

$$\text{NTUmin}(\mathbf{c}) = (|S| - 1)c_{R_2} + c_{R_1}. \tag{3.15}$$

Case 4: If $|S_1| > 1$ *then, letting* ℓ_2 *be the agent from* $\mathcal{E}_1 \setminus S$ *that has second-highest cost,*

$$\text{NTUmin}(\mathbf{c}) = |S_2|c_{R_2} + \min(c_{R_1}, c_{R_2} - c_{\ell_2}) + |S_1 - 1| \min(c_{R_1}, c_{R_2} - \min(c_{R_1}, c_{R_2} - c_{\ell_2})). \tag{3.16}$$

Proof. As we have seen, in Lemma 3.11, that Algorithm 1 correctly computes the NTU-min value, we can examine Algorithm 1 to verify that it produces the results given by the cases here. In cases 1 and 2 there are no agents with quantity 1, so running Algorithm 1 will result in the 'if' statement in line 5 being false. For case 1, this is followed by the 'if' statement in line 10 being true — therefore each agent's bid is allocated in line 11 of Algorithm 1, so each agent $e \in S$ has $b_e^{\min} = c_{R_2}$ proving that $b_S^{\min} = |S|c_{R_2}$ and hence NTUmin $(\mathbf{c}) = |S|c_{R_2}$.

Case 2 is similar, but being false in line 10 results in each bid being allocated in line 13, and hence NTUmin $= |S|c_{R_1}$.

For case 3, there is only a single agent e with quantity 1, which will be allocated a bid value in line 6. As here $\ell_2 = \emptyset$ and we will consider that $c_{\ell_2} = 0$, then e will be allocated a bid of c_{R_1} (as $c_{R_1} \leq c_{R_2} - 0$). All other $|S| - 1$ agents (with quantity 2) will be allocated a bid value in line 11 (including any agent with quantity 1 in S implies that $q_S = Q$), and hence NTUmin$(\mathbf{c}) = c_{R_1} + (|S| - 1)c_{R_2}$.

For case 4, we have a single agent with quantity 1 that gets allocated a bid in line 6 - this is $\min(c_{R_1}, c_{R_2} - c_{\ell_2})$, all other agents with quantity 1 are allocated their bid in line 8, this bid is equivalent to $\min(c_{R_1}, c_{R_2} - \min(c_{R_1}, c_{R_2} - c_{\ell_2})$ (as $b_\ell^{\min} = \min(c_{R_1}, c_{R_2} - c_{\ell_2})$). Finally each agent with quantity 2 is allocated a bid in line 10, and summing up the three different bid values gives NTUmin$(\mathbf{c}) = |S_2|c_{R_2} + \min(c_{R_1}, c_{R_2} - c_{\ell_2}) + (|S_1| - 1)\min(c_{R_1}, c_{R_2} - \min(c_{R_1}, c_{R_2} - c_{\ell_2}))$.

□

Simplifying lower bounds for NTUmin

We will now briefly use the proofs in Theorem 3.12 to show some simple lower bounds for NTUmin. These lower bounds are easier to analyse than the complete NTUmin calculations, and will be used in a later section when examining frugality.

Proposition 3.13. *For all instances having $q_S > Q$, the inequality* NTUmin $\geq (|S|c_{R_1})$ *holds.*

Proof. This trivially follows from case 2 in Theorem 3.12. □

Proposition 3.14. *For all instances having $q_S = Q$, the inequality* NTUmin $\geq (|S_2|c_{R_2})$ *holds.*

Proof. This follows from cases 1,3 and 4 in Theorem 3.12. □

Proposition 3.15. NTUmin $\geq \min(|S_1|c_{R_1}, c_{R_2^*})$.

Proof. When $|S_1| = 0$ this trivially holds; when $|S_1| = 1$ then case 3 in Theorem 3.12 applies, and we can use Equation 3.15. This trivially shows that NTUmin$\geq |S_1|c_{R_1}$ and proves the claim for this case. We will now look at the more interesting case, when $|S_1| > 1$.

The first case of this that we will examine is when $c_{R_2} < c_{R_2^*}$ (remember that R_2 is any cheapest subset with quantity 2, and that R_2^* contains only an agent with quantity 2).

Case 1: $c_{R_2} < c_{R_2^*}$

As we know that R_2 is made up of two agents, we will call these j, k where $c_j = c_{R_1}$ and $c_k \geq c_{R_1}$. Hence $c_{R_2} = c_{R_1} + c_k$ We also know that $c_{\ell_2} \leq c_k$ or else k would be in the lowest-cost set, hence $c_{R_2} \geq c_{R_1} + c_{\ell_2}$, which can be rewritten as $c_{R_1} \leq c_{R_2} - c_{\ell_2}$. We can now use this to rewrite Equation 3.16 as

$$\text{NTUmin} = |S_2|c_{R_2} + c_{R_1} + |S_1 - 1| \min(c_{R_1}, c_{R_2} - c_{R_1})$$

and clearly $c_{R_2} \geq c_{R_1} + c_{R_1}$, hence $c_{R_2} - c_{R_1} \geq c_{R_1}$, so this can be simplified to

$$\text{NTUmin} = |S_2|c_{R_2} + c_{R_1} + (|S_1| - 1)c_{R_1}$$

and therefore

$$\text{NTUmin} = |S_2|c_{R_2} + |S_1|c_{R_1}$$

which proves that NTUmin $\geq |S_1|c_{R_1}$ and hence the claim is proven for this case.

Case 2: $c_{R_2} = c_{R_2^*}$

Let $b_\ell^{\min} = \min(c_{R_1}, c_{R_2} - c_{\ell_2})$. This is the bid made by the agent with largest cost in S_1 (from Proposition 3.7) and rewrite Equation 3.16 as

$$\text{NTUmin} = |S_2|c_{R_2} + b_\ell^{\min} + (|S_1| - 1)\min(c_{R_1}, c_{R_2} - b_\ell^{\min}).$$

We will take this as two cases, where the assumption that $c_{R_1} \leq c_{R_2} - b_\ell^{\min}$ either holds or does not.

Case 2.1: $c_{R_1} \leq c_{R_2} - b_\ell^{\min}$

We know that $b_\ell^{\min} \geq c_\ell \geq c_{\ell_2}$, hence it also follows by transitivity that $c_{R_1} \leq c_{R_2} - c_{\ell_2}$, so we can rewrite Equation 3.16 as

$$\text{NTUmin} \geq |S_2|c_{R_2} + c_{R_1} + (|S_1| - 1)c_{R_1}$$

and as

$$\text{NTUmin} \geq |S_2|c_{R_2} + |S_1|c_{R_1}$$

hence

$$\text{NTUmin} \geq |S_1|c_{R_1}$$

and the claim is proven for this case.

Case 2.2: $c_{R_1} > c_{R_2} - b_\ell^{\min}$

We can take a lower bound on Equation 3.16, using $|S_1| > 1$

$$\text{NTUmin} \geq |S_2|c_{R_2} + b_\ell^{\min} + \min(c_{R_1}, c_{R_2} - b_\ell^{\min})$$

and use our assumption in this case 2.2 ($c_{R_1} > c_{R_2} - b_\ell^{\min}$) to simplify to

$$\text{NTUmin} \geq |S_2|c_{R_2} + b_\ell^{\min} + c_{R_2} - b_\ell^{\min}$$

and hence

$$\text{NTUmin} \geq |S_2|c_{R_2} + c_{R_2}.$$

As case 2 also assumes that $c_{R_2} = c_{R_2^*}$ then we can observe that NTUmin $\geq c_{R_2^*}$ and hence the claim is proven.

□

3.4 The $\alpha\mathcal{M}$ mechanism

We have seen that VCG has a frugality ratio of $\Omega(Q)$, so we now propose a mechanism, $\alpha\mathcal{M}$, that has a much lower frugality ratio ($\Omega(\sqrt{Q})$). We study here, the $\alpha\mathcal{M}$ mechanism when $\alpha = \sqrt{Q}$, and aim to show an upper bound on the frugality for this mechanism. We will do this by partitioning S^α, the winning set under the $\alpha\mathcal{M}$ mechanism, into three subsets — they are $S^\alpha \cap S_1$,$S^\alpha \cap S_2$, and $S^\alpha \setminus S$. We will examine the payments made to each of the subsets, and then combine these payments to give an upper bound for S^α. Recall that, when $\alpha = 1$ the $\alpha\mathcal{M}$ mechanism is equivalent to VCG and we have seen $\phi_{\text{NTUmin}}(\text{VCG}) \geq Q$ so we will be looking to achieve a lower frugality ratio than Q.

3.4.1 Frugality results for the $\alpha\mathcal{M}$ mechanism

We first see some technical propositions that give a bound on the payment value to an agent, based on the cost of some set of agents that could replace it in the winning set. These are given as separate propositions to try and keep the notation simple, but the proofs are identical save for the variation in which of the agents have their bids scaled by α and which do not.

Proposition 3.16. *For all instances of $\alpha\mathcal{M}$, for all $e \in S^\alpha$ and any set $V \subseteq (\mathcal{E} \setminus S^\alpha)$ with $q_{S^\alpha} - q_e + q_V \geq Q$ the inequality $p_e \leq c_V\alpha$ holds.*

Proof. Observe that $q_{S^\alpha} - q_e + q_V \geq Q$ implies that $S^\alpha \setminus \{e\} \cup V$ is a feasible set, and hence that the payment made to e is upper bounded by the maximum that e could bid yet still be chosen in preference to V.

Assume for contradiction that $p_e > c_V\alpha$; as the mechanism pays threshold payments, then $p_e > c_V\alpha$ implies that agent e could possibly be chosen with a threshold bid $b_e = p_e$. Hence if $p_e > c_V\alpha$ then e is still chosen by the $\alpha\mathcal{M}$ mechanism when we have

$$b_e > c_V\alpha. \tag{3.17}$$

The mechanism chooses the winning set by comparing the virtual bids, and as both e and V may be added to $S^\alpha \setminus \{e\}$ to make a feasible set, and e was chosen, this implies that $v_e \leq v_V$. From the definition of the mechanism, we see that $v_V \leq b_V\alpha$ and that $v_e \geq b_e$. (As the mechanism is truthful we can substitute c_V, c_e for b_V, b_e). Therefore if $v_e \leq v_V$ and $v_e \geq b_e$ then through transitivity $b_e \leq v_V$. Again, through transitivity, with $v_V \leq b_V\alpha$, then we have

$$b_e \leq b_V\alpha$$

and

$$b_e \leq c_V\alpha$$

giving a contradiction with 3.17. □

Recall the notation that $S_1^\alpha = S^\alpha \cup \mathcal{E}_1$ and $S_2^\alpha = S^\alpha \cup \mathcal{E}_2$, i.e. S_1^α are the agents in the winning set S^α that have quantity 1, and S_2^α are the agents in S^α with quantity 2.

Proposition 3.17. *For all instances of $\alpha\mathcal{M}$, for all $e \in S_1^\alpha$ and any set $V \subseteq (\mathcal{E} \setminus S^\alpha)$ the inequality $p_e \leq c_V$ holds.*

Proof. Assume for contradiction that $p_e > c_V$; as the mechanism pays threshold payments, then $p_e > c_V$ implies that agent e could possibly be chosen with a threshold bid $b_e = p_e$. Hence if $p_e > c_V$ then e is still chosen by the $\alpha\mathcal{M}$ mechanism when we have

$$b_e > c_V. \tag{3.18}$$

The mechanism chooses the winning set by comparing the virtual bids, and as both e and V may be added to $S^\alpha \setminus \{e\}$ to make a feasible set, and e was chosen, this implies that $v_e \leq v_V$. From the definition of the mechanism, we see that $v_V \leq b_V\alpha$ and that $v_e = b_e\alpha$. (As the mechanism is truthful we can substitute c_V, c_e for b_V, b_e). Therefore if $v_e \leq v_V$ and $v_e \geq b_e\alpha$ then through transitivity $b_e\alpha \leq v_V$. Again, through transitivity, with $v_V \leq b_V\alpha$, then we have $b_e\alpha \leq b_V\alpha$ hence

$$b_e \leq b_V$$

and

$$b_e \leq c_V$$

giving a contradiction with Inequality 3.18. □

Proposition 3.18. *For all instances of $\alpha\mathcal{M}$, for all $e \in S_1^\alpha$ and any $V \subseteq (\mathcal{E}_2 \setminus S^\alpha)$ the inequality $p_e \leq c_V/\alpha$ holds.*

Proof. Assume for contradiction that $p_e > c_V/\alpha$; as the mechanism pays threshold payments, then $p_e > c_V/\alpha$ implies that agent e could possibly be chosen with a threshold bid $b_e = p_e$. Hence if $p_e > c_V/\alpha$ then e is still chosen by the $\alpha\mathcal{M}$ mechanism when we have

$$b_e > c_V/\alpha. \tag{3.19}$$

The mechanism chooses the winning set by comparing the virtual bids, and as both e and V may be added to $S^\alpha \setminus \{e\}$ to make a feasible set, and e was chosen, this implies that $v_e \leq v_V$. From the definition of the mechanism, we see that $v_V = b_V$ and that $v_e = b_e\alpha$. (As the mechanism is truthful we can substitute c_V, c_e for b_V, b_e.) Therefore if $v_e \leq v_V$ and $v_e = b_e\alpha$ then through substitution $b_e\alpha \leq v_V$. Again, through transitivity, with $v_V = b_V$, then we have $b_e\alpha \leq b_V$ and rewriting gives

$$b_e \leq b_V/\alpha$$

and

$$b_e \leq c_V/\alpha$$

giving a contradiction with 3.19. □

We now have one more technical proposition, which tells us which agents can appear in $(S^\alpha \setminus S)$ and $(S \setminus S^\alpha)$. Informally, no agent with quantity 2 will be removed from an optimal set S, and no agent with quantity 1 will be introduced.

Proposition 3.19. *For all instances of $\alpha\mathcal{M}$, having winning set S^α, $(S^\alpha \setminus S) \cap \mathcal{E}_2 = \emptyset$ and $(S \setminus S^\alpha) \cap \mathcal{E}_1 = \emptyset$.*

Proof. Firstly, for contradiction, assume that $\exists e \in ((S^\alpha \setminus S) \cap \mathcal{E}_2)$. Let $V_e \subseteq S \setminus S^\alpha$ be the lexicographically first subset with smallest quantity such that $q_S - q_e + q_{V_e} \geq Q$.

Without loss of generality, assume that e is before V_e lexicographically. As $V_e \subseteq S$ and $e \notin S$ then $c_e < c_{V_e}$ (this is strict due to the lexicographical ordering assumed). Conversely, as $e \in S^\alpha$ and $V_e \cap S^\alpha = \emptyset$ then we have $v_e \leq v_{V_e}$ as e was chosen by the mechanism in preference to V_e, based on their virtual bids.

From the mechanism definition, and assuming truthfulness, we have $v_e = c_e$ and $v_{V_e} \geq c_{V_e}$, hence we get $c_e \geq c_{V_e}$ giving a contradiction with $c_e < c_{V_e}$, showing that there is no $e \in ((S^\alpha \setminus S) \cap \mathcal{E}_2$ and therefore $(S^\alpha \setminus S) \cap \mathcal{E}_2 = \emptyset$. Observe that assuming V_e is before e lexicographically simply gives $c_e \leq c_{V_e}$ and $c_e > c_{V_e}$ instead.

As we have seen that $\forall e \in (S^\alpha \setminus S)$ the equation $q_e = 1$ holds, it is simple to see that there is no $j \in (S \setminus S^\alpha)$ with $q_j = 1$, as the same agent with quantity 1 must be preferred by both the choice for S and the choice for S^α.

□

Now we move on to the body of the proof, considering the partition of the winning set S^α into three sets ($S^\alpha \cap S_1$, $S^\alpha \cap S_2$, and $S^\alpha \setminus S$). We will show payment bounds of $\alpha\mathcal{M}$ for these sets.

Lemma 3.20. *For every instance of $\alpha\mathcal{M}$ where $\alpha = \sqrt{Q}$, the inequality $p_{S^\alpha \cap S_1} \leq \sqrt{Q}$NTUmin holds.*

Proof. We will prove this lemma as two cases. Recalling the definition of NTUmin, given in Definition 1.2. Condition (3) states that

$$\text{for every } e \in S, \text{ there is } T_e \in \mathcal{F} \text{ such that } e \notin T_i \text{ and } \sum_{e' \in S \setminus T_e} b_{e'} = \sum_{e' \in T_e \setminus S} c_{e'}.$$

Case 1: For every $e \in S_1^\alpha$ there exists a T_e set satisfying (3) when $(T_e \setminus S) \cap \mathcal{E}_1$ is not empty.

From the case definition, there is some agent $j \in T_e \setminus S$ such that $q_j = 1$. Applying Proposition 3.2 or 3.3 repeatedly shows that there exists a T'_e such that $q_{T'_e \setminus S} = 1$. We can let $T'_e = S \setminus \{e\} \cup \{j\}$ and hence we have $b_e^{\min} = c_j$, giving $b_e^{\min} = c_{R_1}$ (from R_1 having minimal cost).

As e has quantity 1, it can be replaced by any agent, and Proposition 3.17 gives $p_e \leq c_j$. As we have $b_e^{\min} = c_{R_1}$, then through substitution we have $p_e \leq b_e^{\min}$, and summing over $S^\alpha \cap S_1$ gives

$$p_{S^\alpha \cap S_1} \leq b_{S^\alpha \cap S_1}^{\min}$$

and

$$p_{S^\alpha \cap S_1} \leq \text{NTUmin}.$$

Case 2: Otherwise (i.e. for some $e \in S_1^\alpha$ there is some T_e set satisfying condition (3) in Definition 1.2 when $(T_e \setminus S) \cap \mathcal{E}_1$ is empty).

From Proposition 3.2 we must have some T'_e such that $q_{T'_e \setminus S} = 2$ (as $T_e \setminus S$ contains only agents with quantity 2). As we have seen $q_S = Q$ then we must have $q_{S \setminus T'_e} \leq q_{T'_e \setminus S}$ (or else we would have $q_{T'_e \setminus S} < q_{S \setminus T'_e}$ and $q_{T'_e} < Q$).

Fix $j \in T'_e \setminus S$ to be any agent with $q_j = 2$. By satisfying condition (3) in Definition 1.2 there exists some $S \setminus T'_e$ such that $b_{S \setminus T'_e}^{\min} = c_{T'_e \setminus S}$.

As $S \setminus T'_e \subseteq S$ we have $b_S^{\min} \geq c_j$ and hence

$$\text{NTUmin} \geq c_j. \tag{3.20}$$

We will know examine the payments to each $e \in (S^\alpha \cap S_1)$ as two sub-cases.

Case 2.1: $j \notin S^\alpha$.

From Proposition 3.18 we have $p_e \leq c_j/\alpha$ and summing up over $S^\alpha \cap S_1$ gives $p_{S^\alpha \cap S_1} \leq |S^\alpha \cap S_1| c_j/\alpha$. As we have defined $\alpha = \sqrt{Q}$ and we can observe that $|S^\alpha \cap S_1| \leq Q$ then we can express this as $p_{S^\alpha \cap S_1} \leq Qc_j/\sqrt{Q}$ and hence $p_{S^\alpha \cap S_1} \leq \sqrt{Q}c_j$.

Therefore, with Inequality 3.20, we have

$$p_{S^\alpha \cap S_1} \leq \sqrt{Q}\text{NTUmin}.$$

Case 2.2: $j \in S^\alpha$.

If we examine Lemmas 3.6,3.8, and 3.9 we see that every agent $e \in S_1$ must make an NTUmin bid which is one of at most two distinct values, and if these two values are different then only one agent may bid the higher amount. Let these two bid values be b^y, b^z and assume that $b^z \geq b^y$ (if only one value exists, assume for simplicity that $b^z = b^y$). Hence we can describe the NTUmin bids by

$$b_{S_1}^{\min} = (|S_1| - 1)b^y + b^z. \tag{3.21}$$

Recall that $\ell_1 \in \text{argmax}_{i \in S_1} c_i$ is the lexicographically first agent with the highest cost, and that $\ell_2 \in \text{argmax}_{i \in S_1 \setminus \{\ell_1\}} c_i$ is the lexicographically first agent with second highest cost.

We can observe, from condition (1) in Definition 1.2, that $b^z \geq c_{\ell_1}$ and that when $|S_1| > 1$ then ℓ_2 exists and $b^y \geq c_{\ell_2}$.

Hence we can substitute in Equation 3.21 to get

$$b_{S_1}^{\min} \geq (|S_1| - 1)c_{\ell_2} + c_{\ell_1}. \tag{3.22}$$

If $|S_1| = 1$ then we simply have $p_e \leq b_{S \setminus S^\alpha}^{\min}$ and hence $p_e \leq$ NTUmin giving $p_{S^\alpha \cap S_1} \leq$ NTUmin.

Now we will consider when $|S_1| > 1$.

We can firstly observe that $q_S = q_{S^\alpha} = Q$, as $q_S = Q$ is clear from S being minimal and S_1 being non-empty. As $|S_1| > 1$ then for any size of $S^\alpha \setminus S$ there is some subset of equal quantity in S that can be removed so that $q_{S \setminus S^\alpha} = q_{S^\alpha \setminus S}$ and hence $q_{S^\alpha} = q_S$ (as S^α is minimal with respect to set inclusion).

Therefore $q_{S \setminus S^\alpha} \geq q_{S^\alpha \setminus S} \geq 2$, and as j was not chosen by S, we must assume that $S \setminus S^\alpha$ must contain at least two agents with quantity 1. As ℓ_1, ℓ_2 have the highest cost (and hence bids as $\alpha\mathcal{M}$ was shown to be truthful in Section 3.3.2), we can assume that $\ell_1, \ell_2 \in S \setminus S^\alpha$.

As neither ℓ_1 or ℓ_2 will be chosen in S^α, and $c_{\ell_2} \leq c_{\ell_1}$ we can use Proposition 3.17 to give $p_e \leq c_{\ell_2}$.

Summing up over all $e \in S^\alpha \cap S_1$ gives

$$p_{S^\alpha \cap S_1} \leq |S^\alpha \cap S_1| c_{\ell_2}.$$

We have $|S^\alpha \cap S_1| < |S_1|$ (as we saw that $(S \setminus S^\alpha) \cap S_1$ contains at least two agents) hence we can upper bound with

$$p_{S^\alpha \cap S_1} \leq (|S_1| - 1)c_{\ell_2}$$

Clearly, as $c_{\ell_1} \geq 0$, we have $(|S_1|-1)c_{\ell_2} \leq (|S_1|-1)c_{\ell_2} + c_{\ell_1}$ so we can use the lower bound for $b^{\min}_{S_1}$ given in Inequality 3.22 to get

$$p_{S^\alpha \cap S_1} \leq b^{\min}_{S_1}$$

and hence

$$p_{S^\alpha \cap S_1} \leq \text{NTUmin}.$$

□

Lemma 3.21. *For every instance of $\alpha\mathcal{M}$ having $\alpha = \sqrt{Q}$ the inequality $p_{S^\alpha \cap S_2} \leq b^{\min}_{S^\alpha \cap S_2}\sqrt{Q}$ holds.*

Proof. Recall the definition of set R_2, which was given in Algorithm 1, as the lowest-cost subset from $\mathcal{E} \setminus S$ with quantity at least 2. We will examine the proof as two cases; firstly when the replacement set R_2 was not chosen in S^α by $\alpha\mathcal{M}$.

Case 1 : $R_2 \subseteq \mathcal{E} \setminus S^\alpha$.

Let e be some agent in $S^\alpha \cap S_2$. As $q_{R_2} = q_e = 2$ then R_2 could replace e in S^α to make a feasible set; hence $p_e \leq c_{R_2}\alpha$, from Proposition 3.16. Observe that when $S_1 \neq \emptyset$ holds then $q_S = Q$, and from Lemma 3.4 we have $b^{\min}_e = c_{R_2}$. By substitution this gives $p_e \leq b^{\min}_e \alpha$. Summing this up over every e in $S^\alpha \cap S_2$ gives

$$p_{S^\alpha \cap S_2} \leq b^{\min}_{S^\alpha \cap S_2}\alpha.$$

The second case is when the replacement set R_2 was chosen in the winning set S^α by $\alpha\mathcal{M}$.

Case 2 : $R_2 \subseteq S^\alpha$.

Let e be some agent in $S^\alpha \cap S_2$. Let $R_2^\alpha \subseteq S^\alpha \setminus S$ be a subset such that $q_{S^\alpha \setminus \{e\} \cup R_2^\alpha} \geq Q$. (i.e. $q_{R_2^\alpha} = 1$ if $q_{S^\alpha} > Q$ and $q_{R_2^\alpha} = 2$ where $q_{S^\alpha} = Q$).

As the lowest-cost winning set S included R_2^α in preference to R_2 then we have $c_{R_2^\alpha} \leq c_{R_2}$. From Proposition 3.16 we have $p_e \leq c_{R_2^\alpha}\alpha$ and hence $p_e \leq c_{R_2}\alpha$. from Lemma 3.4 we have $b^{\min}_e = c_{R_2}$. Hence, by substitution, this gives $p_e \leq b^{\min}_e\alpha$. Summing this up over every e in $S^\alpha \cap S_2$ gives

$$p_{S^\alpha \cap S_2} \leq b^{\min}_{S^\alpha \cap S_2}\alpha.$$

□

Lemma 3.22. *For every instance of $\alpha\mathcal{M}$ when $\alpha = \sqrt{Q}$ the inequality $p_{S^\alpha \setminus S} \leq \sqrt{Q}b^{\min}_{S \setminus S^\alpha}$ holds.*

Proof. Firstly, from Proposition 3.19 when $S^\alpha \backslash S$ is non-empty then S_1 is also non-empty (the lemma trivially holds if $S^\alpha \setminus S$ is empty).

We will now consider the more simple case when $q_{S^\alpha} > Q$.

Case 1: $q_{S^\alpha} > Q$

As (from Proposition 3.19) there is some agent $i \in S \setminus S^\alpha$ with $q_i = 1$ then agent $e \in (S^\alpha \backslash S)$ could be replaced by agent $i \in (S \backslash S^\alpha)$ as $q_{S^\alpha} - q_e + q_i = Q+1-2+1 = Q$.

Let $i \in \operatorname{argmin}_{j \in S \backslash S^\alpha} c_j$ and from Proposition 3.16 we have $p_e \le c_i \alpha$. Sum up over $e \in S^\alpha \setminus S$ to get

$$p_{S^\alpha \backslash S} \le |S^\alpha \setminus S| c_j \alpha.$$

Observe (using Proposition 3.19) that $|S^\alpha \setminus S| \le |S \setminus S^\alpha|$ and we get

$$p_{S^\alpha \backslash S} \le |S \setminus S^\alpha| c_j \alpha$$

and as j has the minimum cost in $S \setminus S^\alpha$ then it follows that $|S \setminus S^\alpha| c_j \le c_{S \backslash S^\alpha}$ and hence

$$p_{S^\alpha \backslash S} \le c_{S \backslash S^\alpha} \alpha$$

and from condition (1) in Definition 1.2

$$p_{S^\alpha \backslash S} \le b^{\min}_{S \backslash S^\alpha} \alpha.$$

Now we consider the remaining case.

Case 2: $q_{S^\alpha} = Q$

As S_1 is non-empty, it follows that $q_S = Q$ also, and so $q_{S \backslash S^\alpha} = q_{S^\alpha \backslash S}$. From Proposition 3.19 we then have $|S \setminus S^\alpha| = 2|S^\alpha \setminus S|$. Let $i, j \in S \setminus S^\alpha$ be the two agents with lowest cost.

Observe that as no two agents costs may sum up to less than $c_i + c_j$ we then have a lower bound on the cost of $S \setminus S^\alpha$, which is $c_{S \backslash S^\alpha} \ge |S \setminus S^\alpha|(c_i + c_j)/2$. Using $|S \setminus S^\alpha| = 2|S^\alpha \setminus S|$ we then have

$$c_{S \backslash S^\alpha} \ge |S^\alpha \setminus S|(c_i + c_j) \tag{3.23}$$

For each $e \in S^\alpha \setminus S$, from Proposition 3.16, we have $p_e \le (c_i + c_j)\alpha$, and summing up gives

$$p_{S^\alpha \backslash S} \le |S^\alpha \setminus S|(c_i + c_j)\alpha$$

and with 3.23

$$p_{S^\alpha \backslash S} \le c_{S \backslash S^\alpha} \alpha$$

and from condition (1) in Definition 1.2

$$p_{S^\alpha \setminus S} \leq b^{\min}_{S \setminus S^\alpha} \alpha$$

which completes the proof.

□

Theorem 3.23. *Suppose that there is a $\{1,2\}$ single-commodity auction requiring quantity Q. Then for the $\alpha\mathcal{M}$ mechanism with $\alpha = \sqrt{Q}$ the frugality ratio is upper bounded with $\phi_{\text{NTUmin}}(\alpha\mathcal{M}) \leq 2\sqrt{Q}$.*

Proof. From Lemmas 3.21, 3.20, and 3.20, we have the following

$$\begin{aligned} p_{S^\alpha \cap S_2} &\leq \sqrt{Q} b^{\min}_{S^\alpha \cap S_2}, \\ p_{S^\alpha \cap S_1} &\leq \sqrt{Q}\text{NTUmin}, \\ p_{S^\alpha \setminus S} &\leq \sqrt{Q} c_{S \setminus S^\alpha}. \end{aligned} \tag{3.24}$$

As $S \setminus S^\alpha$ is obviously disjoint with $S^\alpha \cap S_2$ and both are in S, then $b^{\min}_{S^\alpha \cap S_2} + c_{S \setminus S^\alpha} \leq b^{\min}_S \leq \text{NTUmin}$. Therefore, we have $p_{S^\alpha \cap S_2} + p_{S^\alpha \setminus S} \leq \sqrt{Q}\text{NTUmin}$, and we can add this to Inequality 3.24 to give

$$p_S \leq 2\sqrt{Q}\text{NTUmin}.$$

This shows that the payment is upper bounded by $2\sqrt{Q}\text{NTUmin}$ and hence we have

$$\phi_{\text{NTUmin}}(\alpha\mathcal{M}) \leq 2\sqrt{Q}.$$

□

We will now briefly consider the upper bound with respect to NTUmax.

Corollary 3.24. *For $\{1,2\}$ Single-Commodity Auctions with quantity Q, the $\alpha\mathcal{M}$ scaling mechanism when $\alpha = \sqrt{Q}$, the frugality ratio is upper bounded with $\phi_{\text{NTUmax}}(\alpha\mathcal{M}) \leq 2\sqrt{Q}$.*

Proof. From their definitions, NTUmax $\geq$ NTUmin, hence the proof in Theorem 3.23 trivially applies with regard to NTUmax.

□

3.4.2 A lower bound on frugality with the $\alpha\mathcal{M}$ mechanism

We will show lower bounds by using a series of examples. For any instance of a set-system auction let the total payment made to all agents be $p_\mathcal{E}$ and let the *payment ratio* be $\frac{p_\mathcal{E}}{\text{NTUmin}}$. Hence the frugality ratio for a mechanism is given by the maximum payment ratio over all possible instances. This means that, when we have a mechanism that has an instance with some payment ratio $\frac{p_{\mathcal{E}}}{\text{NTUmin}}$ then $\frac{p_{\mathcal{E}}}{\text{NTUmin}}$ is a lower bound for the frugality ratio of that mechanism.

Recall that when $\alpha = 1$ this mechanism is exactly equivalent to VCG (no scaling takes place, and all agents are paid their threshold value). Recall that Example 3.1 showed an instance I where the frugality ratio is $\phi_{\text{NTUmin}}(\alpha\mathcal{M}) \geq Q$. As this shows a small value of α does not improve frugality, perhaps some larger value of α may produce better results. Example 3.2 shows the same instance I with the $\alpha\mathcal{M}$ mechanism when $\alpha = Q$.

Example 3.2. *In this example we have a commodity auction for quantity Q items and observe that the number of agents $n = Q + 1$. For each agent $e \in \mathcal{E}$ the quantity q_e and cost c_e are given in Table 3.2. A value $b_e^{\min}$ for a* NTUmin *bid vector is also given, as is the payment made by the $\alpha\mathcal{M}$ mechanism p_e^{α}.*

Agent	q_e	c_e	$b_e^{\min}$	p_e^{α}
A_1	1	0	1	$1/Q$
A_2	1	0	0	$1/Q$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_{n-1}	1	0	0	$1/Q$
A_n	2	1		

TABLE 3.2: Instance I with $\alpha = Q$

To verify that the payments given are correct, observe that $v_n = c_n = 1$, and when there is a threshold bid $b_e = 1/Q$ then applying the scaling for $e < n$ gives $v_e = Q(1/Q)$ so clearly $v_e = 1 = v_n$ making $p_e = 1/Q$ the threshold value. Now for instance I we would have a payment of $1/Q$ to all $n-1$ winning agents, giving a payment ratio of $\frac{Q}{Q} = 1$.

While Example 3.2 having a frugality ratio of 1 is obviously a positive result, we now examine, in Example 3.3, another instance of the auction, I' where we see that $\alpha = Q$ gives poor frugality.

Example 3.3.

In this example we have a commodity auction for quantity Q items and observe that the number of agents $n = Q/2 + 2$. For each agent $e \in \mathcal{E}$ the quantity q_e and cost c_e are given in Table 3.3. A value $b_e^{\min}$ for a NTUmin *bid vector is also given, as is the payment made by the $\alpha\mathcal{M}$ mechanism p_e^{α}.*

Agent	q_e	c_e	$b_e^{\min}$	p_e^{α}
A_1	2	0	2	$2Q$
A_2	2	0	2	$2Q$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_{n-2}	2	0	2	$2Q$
A_{n-1}	1	1		
A_n	1	1		

TABLE 3.3: Instance I' with $\alpha = Q$

Here, there is a payment of $2Q$ *to all* $n-2$ *winning agents, (when the threshold bid* $b_e = 2Q$ *then* $v_e = 2Q = (v_{n-1} + v_n)$ *making* $p_e = 2Q$ *a threshold value). This gives a payment ratio of* Q*.*

Example 3.3 shows that $\alpha\mathcal{M}$ has a frugality ratio that is no lower than that of VCG, which was also shown to be Q in Example 3.1.

Although we can see that Instance I' does give a small payment ratio for $\alpha\mathcal{M}$ when $\alpha = Q$.

Example 3.4. *In this example we have a commodity auction for quantity* Q *items and observe that the number of agents* $n = Q/2 + 2$*. For each agent* $e \in \mathcal{E}$ *the quantity* q_e *and cost* c_e *are given in Table 3.4. A value* $b_e^{\min}$ *for a* NTUmin *bid vector is also given, as is the payment made by the VCG mechanism* p_e^{VCG}*.*

Agent	q_e	c_e	$b_e^{\min}$	p_e^{VCG}
A_1	2	0	2	2
A_2	2	0	2	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_{n-2}	2	0	2	2
A_{n-1}	1	1		
A_n	1	1		

TABLE 3.4: Instance I' with $\alpha = 1$ (equivalent to VCG)

Here we see that $\{A_1, \ldots, A_{n-2}\}$ *is a winning set, the bid vector* $\mathbf{b}^{\min}$ *meets the conditions for a* NTUmin *vector, hence* $\mathrm{NTUmin}(\mathbf{c}) = 2(n-2)$*. For this instance, then, the payment ratio for* $1\mathcal{M}$ *(i.e. VCG) is simply* 1*, which we already know is optimal.*

Calculation of a lower bound

The examples have shown that some instances give low frugality with large values of α, but some other instances only give low frugality with small values of α. We now generalize these examples to show possible ratios of Q/α or α and hence that any scaling mechanism of this type must have a frugality ratio of at least $\sqrt{Q}$.

Theorem 3.25. *For $\{1,2\}$ Single-Commodity Auctions with quantity Q, the $\alpha\mathcal{M}$ scaling mechanism has a frugality ratio given by $\phi_{\text{NTUmin}}(\alpha\mathcal{M}) \geq \sqrt{Q}$ for all values of α.*

Proof. We will now consider both the example instances I and I' under the more general $\alpha\mathcal{M}$ mechanism. By generalizing instance I in Example 3.2 for any value of α we get Example 3.5. Similarly, by generalizing I' in Example 3.3 we get Example 3.6. Let m be some integer parameter that denotes the size of the instance. Note that for all $\alpha > 0$ then the winning sets S^α, S'^α will not change for either instance. We will not consider the mechanism when $\alpha = 0$, as there are no defined threshold payments.

Firstly, examining the instance I in Example 3.5; for each winning agent $e \in S^\alpha$, we can observe that $p_e^\alpha = 1/\alpha$. As agent A_{m+1} could replace agent e to make a feasible set, the threshold payment is upper bounded by the maximum amount that e could bid, and still possibly be chosen in preference to agent A_{m+1}. That is when $v_e = v_{m+1}$, and as we can see that $v_{m+1} = 1$ in this instance, then when $b_e = 1/\alpha$ we have $v_e = 1$ hence $1/\alpha$ is the threshold value.

This gives Q/α as the sum of such payments over all agents in S^α. The example shows a feasible bid vector, $\mathbf{b}^{\min}$ which has $b_S^{\min} = 1$, and hence the NTUmin value for this instance is at most 1 ($\text{NTUmin}(\mathbf{c}) \leq 1$). This gives a payment ratio of $p_{\mathcal{E}}^\alpha/\text{NTUmin}(\mathbf{c}) \geq Q/\alpha$.

Now examining the instance I' in Example 3.6; for each winning agent $e \in S^\alpha$, we can calculate that $p_e = 2\alpha$. As before, we look for a replacement for agent e; in this instance it is the set $\{A_{m+1}, A_{m+2}\}$. Observe that $v_{\{A_{m+1},A_{m+2}\}} = 2\alpha$ and when the threshold bid is $b_e = 2\alpha$ then $v_e = 2\alpha$, which gives $b_e = 2\alpha$ as the threshold value, and hence $p_e^\alpha = 2\alpha$ This gives a total payment of $p_{\mathcal{E}}^\alpha = m2\alpha = Q\alpha$. The bid vector $\mathbf{b}^{\min}$ shows that $\text{NTUmin}(\mathbf{c}) \leq Q$; combining this with the payments shown gives a payment ratio for this instance which is $p_{\mathcal{E}}^\alpha/\text{NTUmin}(\mathbf{c}) \geq \alpha$.

For Instance I we have a payment ratio of at least Q/α, and for I' a payment ratio of at least α. As any $\alpha\mathcal{M}$ mechanism may be applied to both instances I and I' we can see that

$$\phi_{\text{NTUmin}}(\alpha\mathcal{M}) \geq \max\left(\frac{Q}{\alpha}, \alpha\right)$$

hence we can see a lower bound of frugality exists, for any $\alpha\mathcal{M}$ mechanism, of

$$\phi_{\text{NTUmin}}(\alpha\mathcal{M}) \geq \sqrt{Q}.$$

□

Example 3.5. *In this example we have a commodity auction for quantity Q items and observe that the number of agents in the winning set $m = Q$. For each agent $e \in \mathcal{E}$ the quantity q_e and cost c_e are given in Table 3.5. A value $b_e^{\min}$ for a* NTUmin *bid vector is also given, as is the payment made by the $\alpha\mathcal{M}$ mechanism p_e^α.*

	Agent	q_e	c_e	$b_e^{\min}$	p_e^{α}
	A_1	1	1	1	$1/\alpha$
	A_2	1	0	0	$1/\alpha$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	A_m	1	0	0	$1/\alpha$
	A_{m+1}	2	1		
Total				1	m/α
=				1	Q/α

TABLE 3.5: Instance I with $\alpha\mathcal{M}$

Example 3.6. *In this example we have a commodity auction for quantity Q items and observe that the number of agents in the winning set $m = Q/2$. For each agent $e \in \mathcal{E}$ the quantity q_e and cost c_e are given in Table 3.6. A value $b_e^{\min}$ for a* NTUmin *bid vector is also given, as is the payment made by the $\alpha\mathcal{M}$ mechanism p_e^{α}.*

	Agent	q_e	c_e	$b_e^{\min}$	p_e^{α}
	A_1	2	0	2	2α
	A_2	2	0	2	2α
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	A_m	2	0	2	2α
	A_{m+1}	1	1		
	A_{m+2}	1	1		
Total				$2m$	$2m\alpha$
=				Q	$Q\alpha$

TABLE 3.6: Instance I' with $\alpha\mathcal{M}$

3.5 The Unrestricted Integer Single-Commodity Auction

We now turn our attention to the general case for integer single-commodity auctions, and remove the restriction on the quantity that each agent may supply.

3.5.1 A lower bound on frugality for all Scaling Mechanisms

We have seen that our simple scaling mechanism works well for the $\{1, 2\}$ case, so we now consider the frugality of similar scaling mechanisms in the unrestricted case.

Preliminaries

We will now extend the lower bound results from the $\{1, 2\}$ integer auction to show a lower bound for the more general case of the Integer Single-Commodity Auction (which satisfies $\forall e \in \mathcal{E}, q_e \in \mathbb{Z}^+$) with scaling mechanisms similar to those seen in the $\{1, 2\}$ integer auction.

Let β be a scaling function, returning a linear scaling vector, $\mathbf{a} = \beta(Q, k)$ (with $\alpha_e \in \mathbb{R}$). The two parameters of β are Q, which is given in the auction instance, and $k \in \mathbb{Z}$ which is the 'maximum quantity' parameter for any agent, i.e. $\forall e \in \mathcal{E}, q_e \leq k$.

For ease of notation we will assume for the rest of this section that we are using the mechanism $\beta\mathcal{M}$, and hence both $S = S^{\beta\mathcal{M}}$ and $\mathbf{p} = \mathbf{p}^{\beta\mathcal{M}}$ can be assumed. Let $\beta\mathcal{M}$ be the mechanism that uses the scaling vector $\mathbf{a} = (a_1, \ldots, a_k)$ returned by β, as follows.

Compute a 'virtual' bid v_e for each agent e as

$$v_e = b_e a_{q_e}.$$

Let the winning set, $S \in \mathcal{F}$, be a feasible set with the lowest 'virtual' cost, i.e.

$$S \in \operatorname*{argmax}_{T \in \mathcal{F}} v_T.$$

Each agent e will be paid its threshold value, p_e.

If we consider every scaling function β, and the resulting class of mechanisms, then we can think of $\beta\mathcal{M}$ as the class of all 'blind-scaling' mechanisms; when the mechanism must choose a scaling factor for each possible quantity, based only on the quantity required Q and the maximum quantity parameter k (so the mechanism does not look at any more details of each instance before deciding the scaling factors).

Proof of lower bound

The proof will be given by examining a series of example instances given, and showing that at least one of them must cause a payment ratio that satisfies the lower bound. We will begin this, simply by reminding ourselves of Example 3.5, which is duplicated here as Instance I^1 in Example 3.7.

Example 3.7. *In this example we have a* $\{1, 2\}$ *commodity auction for quantity* Q *items. Let* $m = Q$ *and observe that the winning set is given by* $S = \{A_1, \ldots, A_m\}$ *. For each agent* $e \in \mathcal{E}$ *the quantity* q_e *and cost* c_e *are given in Table 3.7. A value* $b_e^{\min}$ *for a* NTUmin *bid vector is also given, giving* NTUmin≤ 1. *The payment made by the* $\beta\mathcal{M}$ *mechanism is also given in the table as* p_e.

	Agent	q_e	c_e	$b_e^{\min}$	p_e
	A_1	1	0	1	$\frac{a_2}{a_1}$
	A_2	1	0	0	$\frac{a_2}{a_1}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
	A_m	1	0	0	$\frac{a_2}{a_1}$
	A_{m+1}	2	1		
Total				1	$Q\frac{a_2}{a_1}$

TABLE 3.7: Instance I^1

Here, there are at least Q agents in S, each is paid $\frac{a_2}{a_1}$, and NTUmin ≤ 1 *(as shown in Section 3.4.2); hence the payment ratio $\frac{p_\mathcal{E}}{\text{NTUmin}} \geq Q\frac{a_2}{a_1}$.*

In order to verify that these payments are correct, we need to establish the threshold bids. For each agent $e \in S$, if $v_e > v_{m+1}$ then agent e would not be chosen, as the winning set would become $S \setminus \{e\} \cup \{m+1\}$. When $v_e = v_{m+1}$, agent e may still be chosen (assuming that it is lexicographically first), hence when agent e bids b_e such that $v_e = v_{m+1}$ this is the threshold bid, and hence the payment made by this mechanism.

As the mechanism is truthful, we can assume $b_{m+1} = c_{m+1}$ hence $v_{m+1} = c_{m+1}a_2 = a_2$. Then, if we assume for all $e \in S$, that $b_e = \frac{a_2}{a_1}$ then as $v_e = b_e a_1$ we have $v_e = \frac{a_2}{a_1}a_1 = a_2 = v_{m+1}$. This shows that $b_e = \frac{a_2}{a_1}$ is a threshold bid for all $e \in S$, hence the payment is given by $p_e = \frac{a_2}{a_1}$, confirming the payment given in Table 3.7.

Example 3.7 illustrates that, in order for the frugality ratio to be less than Q, it is necessary for $a_2 < a_1$. The first technique used in this section will be to see that, in order to meet some given bound on the frugality, the consecutive scaling values (i.e. a_j, a_{j+1}) must be separated by at least a certain ratio. This will be achieved by generalizing Example 3.7, and in order to show how this example will be scaled-up to the general case we will now look at the next example, Example 3.8. This instance, I^2, has agents with quantities 2 and 3.

Example 3.8. *In this example we have a $\{2,3\}$-commodity auction for quantity Q items and observe that the winning set S is given by $S = \{A_1, \dots, A_m\}$ (with $m = \lceil \frac{Q}{2} \rceil$). For each agent $e \in \mathcal{E}$ the quantity q_e and cost c_e are given in Table 3.8. A value $b_e^{\min}$ for a* NTUmin *bid vector is also given, giving* NTUmin≤ 2*. The payment made by the $\beta\mathcal{M}$ mechanism is also given in Table 3.8 as p_e.*

Agent	q_e	c_e	$b_e^{\min}$	p_e
A_1	2	0	1	$\frac{a_3}{a_2}$
A_2	2	0	1	$\frac{a_3}{a_2}$
A_3	2	0	0	$\frac{a_3}{a_2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
A_m	2	0	0	$\frac{a_3}{a_2}$
A_{m+1}	3	1		
A_{m+2}	3	1		

TABLE 3.8: Instance I^2

We can note here that there are now two agents in S *that can have a bid value* $b_e^{\min} = 1$, *and that the vector* $\mathbf{b}^{\min}$ *shows* NTUmin ≤ 2. *(For all* $e \in \{3, \dots, m\}$ *then from* $T = S \setminus \{1, 2, e\} \cup \{m+1, m+2\}$ *then condition (2) in Definition 1.2 gives* $b_1^{\min} + b_2^{\min} + b_e^{\min} \leq c_{m+1} + c_{m+2}$ *giving* $b_e^{\min} \leq 0$*).*

Again, to verify that these payments are correct, we establish the threshold bids. For each agent $e \in S$, *if* $v_e > v_{m+1}$ *then agent* e *would not be chosen, as the winning set could become* $S \setminus \{e\} \cup \{m+1\}$. *When* $v_e = v_{m+1}$, *agent* e *may still be chosen, hence when agent* e *bids* b_e *such that* $v_e = v_{m+1}$ *this is the threshold bid, and hence the payment.*

If we assume for all $e \in S$, *that* $b_e = \frac{a_3}{a_2}$ *then as* $v_e = b_e a_2$ *we have* $v_e = \frac{a_3}{a_2} a_2 = a_3$ *and observe that* $v_{m+1} = a_3$. *This shows that* $b_e = \frac{a_3}{a_2}$ *is a threshold bid and hence the payment is* $p_e = \frac{a_3}{a_2}$.

Here, there are at least $\frac{Q}{2}$ *agents in* S, *each is paid* $\frac{a_3}{a_2}$, *and* NTUmin ≤ 2*; hence the payment ratio*

$$\frac{p_{\mathcal{E}}}{\text{NTUmin}} \geq \frac{Qa_3}{4a_2}$$

Example 3.9 will now generalize these examples. Recall the assumption that k is a maximum size parameter, and that $k \leq \sqrt{Q}$. For each $j \in \{1, \dots, k-1\}$ let Example 3.9 describe instance I^j. Observe the assumption that $j < k \leq \sqrt{Q}$ implies that $m \geq j$ which is required by the structure of the example (m is defined in the example as $m = \lceil \frac{Q}{j} \rceil$).

Example 3.9. *In this example we have a* $\{j, j+1\}$*-commodity auction for quantity* Q *items. Let* $m = \lceil \frac{Q}{j} \rceil$ *and observe that the winning set is given by* $S = \{A_1, \dots, A_m\}$. *For each agent* $e \in \mathcal{E}$ *the quantity* q_e *and cost* c_e *are given in Table 3.9. A value* $b_e^{\min}$ *for a* NTUmin *bid vector is also given, giving* NTUmin$\leq j$. *The payment made by the* $\beta\mathcal{M}$ *mechanism is also given in Table 3.9 as* p_e.

	Agent	q_e	c_e	$b_e^{\min}$	p_e
	A_1	j	0	1	$\frac{a_{j+1}}{a_j}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
	A_j	j	0	1	$\frac{a_{j+1}}{a_j}$
	A_{j+1}	j	0	0	$\frac{a_{j+1}}{a_j}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
	A_m	j	0	0	$\frac{a_{j+1}}{a_j}$
	A_{m+1}	$j+1$	1		
	$\vdots$	$\vdots$	$\vdots$		
	A_{m+j}	$j+1$	1		
Total				j	$m\frac{a_{j+1}}{a_j}$

TABLE 3.9: Instance I^j

Firstly, we can observe that Examples 3.7 and 3.8 are special cases of Example 3.9 with $j = 1$ and $j = 2$ respectively. We can see that with this parameter j, that there are j agents in S that can have a (NTUmin) bid value $b_e = 1$ (for $e \in S$). We can show that there can be no more than j agents that can each bid 1 as follows; $j+1$ agents, each with quantity j, could be 'replaced' by the j agents outside S, each with quantity $j+1$, so no set of $j+1$ agents in S can bid a sum of more than j.

More formally,

$$\forall e \in S, \text{let } T_e = S \setminus \{1, \ldots, j, e\} \cup \{(m+1), \ldots, (m+j+1)\}.$$

Observe that

$$\left(\sum_{i=1}^{j} q_i\right) + q_e = j(j+1)$$

and

$$\left(\sum_{i=1}^{j} +1 q_{m+i}\right) = j(j+1)$$

hence $q_S = q_T$ and T_e is a feasible set.

Using this T_e in condition (3) in Definition 1.2, for all $e \in \{j+1, \ldots, m\}$ gives

$$\left(\sum_{i=1}^{j} b_i^{\min}\right) + b_e^{\min} = \sum_{i=1}^{j} c_{m+1}.$$

As $\sum_{i=1}^{j} b_i^{\min} = j$ and $\sum_{i=1}^{j} c_{m+i} = j$ then we have $b_e^{\min} = 0$, which shows that for all $e \in \{j+1, \dots, m\}$ then vector $\mathbf{b}^{\min}$ has some T_e satisfying condition (3) in Definition 1.2.

For all $e \in \{1, \dots, j\}$, let $T_e = S \setminus \{e\} \cup \{A_{m+1}\}$ which gives $b_e^{\min} = 1$, showing that the bid vector $\mathbf{b}^{\min}$ has, for all $e \in S$, some T_e satisfying condition (3) in Definition 1.2 and as we can observe $\mathbf{b}^{\min}$ satisfies conditions (1) and (2) of Definition 1.2 then this shows NTUmin $\leq b_S^{\min}$ and hence NTUmin $\leq j$.

We can also generalize the payment to each $e \in S$. For each agent $e \in S$, if $v_e > v_{m+1}$ then agent e would not be chosen, as the winning set could become $S \setminus \{e\} \cup (m+1)$. When $v_e = v_{m+1}$, then agent e may still be chosen, hence when agent e bids b_e such that $v_e = v_{m+1}$ this is the threshold bid, and hence the payment.

If we assume for all $e \in S$, that $b_e = \frac{a_{j+1}}{a_j}$ then as $v_e = b_e a_j$ we have $v_e = \frac{a_{j+1}}{a_j} a_j = a_{j+1} = v_{m+1}$. This shows that $b_e = \frac{a_{j+1}}{a_j}$ is a threshold bid for all $e \in S$, hence the payment is given by $p_e = \frac{a_{j+1}}{a_j}$.

Let $\mathbf{c}$ be a cost vector for instance I^j and we examine the payment ratio $\frac{p\mathcal{E}}{\text{NTUmin}}$ as follows; there are at least $\frac{Q}{j}$ agents in S, each is paid $\frac{a_{j+1}}{a_j}$, and NTUmin $\leq j$; hence the payment ratio satisfies the following inequality

$$\frac{p\mathcal{E}}{\text{NTUmin}} \geq \frac{Q a_{j+1}}{j^2 a_j}. \tag{3.25}$$

We can then use this as we move onto the first part of the proof.

We will initially define a 'maximum size' parameter, k, and will examine a series of instances when all agents have quantity at most k. From these instances we will be able to show a lower-bound in terms of this parameter, k. Finally we will show how to compute a value for k that gives a lower-bound for any given Q.

Let $k \in \{1, \dots, \lfloor\sqrt{Q}\rfloor\}$ be a size parameter. and we will consider just the instances $I^1, \dots, I^k$. We will show a minimum ratio needed between any consecutive scaling values (a_j, a_{j+1}) (when $j < k$) in order to satisfy the specific payment ratio given.

Proposition 3.26. *For instance I^j of $\beta\mathcal{M}$ with $j \leq k-1$ and $\frac{a_j}{a_{j+1}} \leq Q^{\frac{1}{k}}$ the inequality $\frac{p\mathcal{E}}{\text{NTUmin}} \geq \frac{Q^{\frac{k-1}{k}}}{k^2}$ holds.*

Proof. From inequality 3.25 we have $\frac{p\mathcal{E}}{\text{NTUmin}} \geq \frac{Q a_{j+1}}{j^2 a_j}$, and as $j \leq k$ implies $\frac{1}{j^2} \geq \frac{1}{k^2}$ and hence $\frac{Q a_{j+1}}{j^2 a_j} \geq \frac{Q a_{j+1}}{k^2 a_j}$ it follows, due to transitivity, that

$$\frac{p\mathcal{E}}{\text{NTUmin}} \geq \frac{Q a_{j+1}}{k^2 a_j}$$

additionally $\frac{a_j}{a_{j+1}} \leq Q^{\frac{1}{k}}$ can be inverted to be expressed as

$$\frac{a_{j+1}}{a_j} \geq Q^{\frac{-1}{k}}$$

therefore, by transitivity

$$\frac{p_{\mathcal{E}}}{\text{NTUmin}} \geq \frac{Q}{k^2}\frac{a_{j+1}}{a_j} \geq \frac{QQ^{\frac{-1}{k}}}{k^2}$$

and as $QQ^{\frac{-1}{k}} = Q^{\frac{k-1}{k}}$ this can be simplified to state

$$\frac{p_{\mathcal{E}}}{\text{NTUmin}} \geq \frac{Q^{\frac{k-1}{k}}}{k^2}.$$

completing the proof.

□

This minimum separation required between every a_j and a_{j+1} implies that there is large separation between a_1 and a_k. We will then see in Example 3.10 that when this large separation between a_1 and a_k exists, this will give rise to a large payment ratio (and hence frugality ratio).

Example 3.10. *In this example we have a commodity auction for quantity Q items with the parameter k. Let $m = \lceil \frac{Q}{k} \rceil$ and observe that the winning set is given by $S = \{A_1, \ldots, A_m\}$. For each agent $e \in \mathcal{E}$ the quantity q_e and cost c_e are given in Table 3.10. A value $b_e^{\min}$ for a* NTUmin *bid vector is also given, giving* NTUmin$\leq mk$. *The payment made by the $\beta\mathcal{M}$ mechanism is also given in the table as p_e.*

Agent	q_e	c_e	$b_e^{\min}$	p_e
A_1	k	0	k	$\frac{ka_1}{a_k}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
A_m	k	0	k	$\frac{ka_1}{a_k}$
A_{m+1}	1	1		
$\vdots$	$\vdots$	$\vdots$		
A_{m+k}	1	1		

TABLE 3.10: Instance I^k

Proposition 3.27. *For instance I^k of $\beta\mathcal{M}$ the inequality $\frac{p_{\mathcal{E}}}{\text{NTUmin}} \geq \frac{a_1}{a_k}$ holds.*

Proof. For each $e \in S$, there is exactly one feasible set not containing e — that is $\mathcal{E} \setminus \{e\}$. Therefore the only bid vector that could satisfy NTUmin must satisfy condition (3) in Definition 1.2 with $T_e = \mathcal{E} \setminus \{e\}$. Therefore the NTUmin bid for each $e \in S$ must be

$$b_e^{\min} = c_{T_e \setminus S} = c_{\{m+1,\ldots,m+k+1\}} = k$$

As there are m agents in S, each having a bid $b_e^{\min} = k$, we have

$$\text{NTUmin} \leq mk.$$

Similarly, the threshold bid for e must be when $v_e = v_{\{m+1,\dots,m+k\}}$. Assuming $b_e = \frac{ka_1}{a_k}$ multiplying by the scaling factor a_k gives $v_e = \frac{ka_1}{a_k}a_k = ka_1$. The virtual bids of the competing agents $i \in \{m+1,\dots,m+k\}$ are $v_i = a_1$, hence $v_{\{m+1,\dots,m+k\}} = ka_1$ showing that

$$b_e = \frac{ka_1}{a_k}$$

is a threshold bid, and hence the payment

$$p_e = \frac{ka_1}{a_k}.$$

Therefore, in Instance I^k, there are m agents in S; each is paid $\frac{ka_1}{a_k}$ giving a total payment of $\frac{mka_1}{a_k}$. As we have seen NTUmin $\leq mk$ hence

$$\frac{p\varepsilon}{\text{NTUmin}} \geq \frac{a_1}{a_k}.$$

$\square$

We will now move on to a result which shows that, for any scaling vector that could be used by the $\beta\mathcal{M}$ mechanism, there will be some instance, from the examples that we have seen, that give a payment ratio of at least $\frac{Q^{\frac{k-1}{k}}}{k^2}$.

Proposition 3.28. *For any scaling vector* a *given by* $\beta\mathcal{M}$ *there is some Instance* I^j *for* $j \in \{1,\dots,k-1\}$ *or Instance* I^k *such that the inequality* $\frac{p\varepsilon}{\text{NTUmin}} \geq \frac{Q^{\frac{k-1}{k}}}{k^2}$ *holds.*

Proof. If there existed some $j \in \{1,\dots,k-1\}$ such that $\frac{a_j}{a_{j+1}} \leq Q^{\frac{1}{k}}$ then Proposition 3.26 implies that $\frac{p\varepsilon}{\text{NTUmin}} \geq \frac{Q^{\frac{k-1}{k}}}{k^2}$. Therefore, if the expression

$$\forall j \in \{1,\dots,k-1\}, \frac{a_j}{a_{j+1}} > Q^{\frac{1}{k}} \tag{3.26}$$

does not hold then the proof is complete.

Hence suppose that expression 3.26 does hold. We can see this implies that the consecutive scaling values must have a certain separation. By way of example, this gives $\frac{a_1}{a_2} > Q^{\frac{1}{k}}$, $\frac{a_2}{a_3} > Q^{\frac{1}{k}}$ etc. By transitivity we would have $\frac{a_1}{a_3} > Q^{\frac{2}{k}}, \frac{a_1}{a_4} > Q^{\frac{3}{k}}$ etc. This can then be generalized, for $j \in 1,\dots,k-1$ to give

$$\frac{a_1}{a_{j+1}} > Q^{\frac{j}{k}}$$

Suppose that $j = k-1$, then we have

$$\frac{a_1}{a_k} > Q^{\frac{k-1}{k}}.$$

Referring back to Proposition 3.27, Instance I^k gives

$$\frac{p\varepsilon}{\text{NTUmin}} \geq \frac{a_1}{a_k}$$

hence, by transitivity,

$$\frac{p\varepsilon}{\text{NTUmin}} > Q^{\frac{k-1}{k}}.$$

As we have seen that, if expression 3.26 does not hold, then the proposition is satisfied for some Instance I_j where $j \in \{1, \ldots, k-1\}$. When expression 3.26 does hold, then we have instance I^k giving $\frac{p\varepsilon}{\text{NTUmin}} \geq \frac{Q^{\frac{k-1}{k}}}{k^2}$, satisfying the proposition. Hence there is some instance, either I^j for $j \in \{1, \ldots, k-1\}$ or I^k that satisfies the proposition. □

Now that we have seen that there is always some instance that gives at least this (reasonably large) payment ratio in terms of k, we can use this to prove a lemma that shows a lower bound on the frugality ratio for all Integer Single-Commodity Auctions.

Lemma 3.29. *For all Integer Single-Commodity Auctions for quantity Q items and a maximum-quantity parameter $k \leq \sqrt{Q}$, every blind-scaling mechanism $\beta\mathcal{M}$ has a frugality ratio which satisfies the inequality $\phi_{\text{NTUmin}}(\beta\mathcal{M}) \geq \frac{Q^{\frac{k-1}{k}}}{k^2}$.*

Proof. The blind-scaling mechanism $\beta\mathcal{M}$ must, by definition, calculate its scaling vector **a** for use on any instance that it may be given with these parameters. Once this scaling vector is fixed the mechanism may possibly be given either Instance I^k or Instance I^j for any $j \in \{1, \ldots, k-1\}$. Proposition 3.28 shows that at least one of these instances gives $\frac{p\varepsilon}{\text{NTUmin}} \geq \frac{Q^{\frac{k-1}{k}}}{k^2}$. The existence of such an instance proves

$$\phi_{\text{NTUmin}}(\beta\mathcal{M}) \geq \frac{Q^{\frac{k-1}{k}}}{k^2}.$$

□

Now that we have shown a lower bound on frugality for values of Q in terms of the parameter k, we can specify a value of k such as to give a lower bound entirely in terms of Q. To that end, suppose $k = \frac{\ln Q}{2}$, and we will see this implies a lower bound of $\frac{4Qe^{-2}}{\ln^2 Q}$ for $\beta\mathcal{M}$ mechanisms.

Theorem 3.30. *Given an Integer Single-Commodity Auction for quantity Q items, every blind-scaling mechanism, $\beta\mathcal{M}$, has a frugality ratio which satisfies the inequality $\phi_{\text{NTUmin}}(\beta\mathcal{M}) \geq \frac{4Qe^{-2}}{\ln^2 Q}$.*

Proof. Considering the proof of Lemma 3.29, suppose $k = \frac{\ln Q}{2}$. Re-arranging the expression given in Lemma 3.29 gives $\frac{Q^{\frac{k-1}{k}}}{k^2} = \frac{4Qe^{-2}}{\ln^2 Q}$, and hence, by substitution

$$\phi_{\text{NTUmin}}(\beta\mathcal{M}) \geq \frac{4Qe^{-2}}{\ln^2 Q}.$$

(Note that this only holds exactly when $\frac{\ln Q}{2} \in \mathbb{Z}$ as k must be an integer parameter but this restriction is not important to the asymptotic result.)

□

Chapter 4

Shortest Path with k-sets

4.1 Overview

One of the most commonly studied types of set-system auctions is the path auction (e.g. [4, 12, 39, 23, 10]). A path auction is a special-case of the set-system auction, where each agent represents an edge in a graph, and the feasible sets are the sets of agents that include a path between two specified vertices of the graph. In this chapter, we propose a generalization of this path auction, and give some preliminary results.

In a typical path auction, we consider that each agent is represented by some edge in a graph (or multigraph), and that the feasible sets are exactly those sets that contain a path between two pre-determined vertices of the graph.

In real-world situations, it is normally the case that, in any given network, not all of the edges are owned by separate entities. While, of course, that does not preclude them being auctioned individually, we will see some reasons why this is not always desirable. Therefore, it seems reasonable to generalize the path auction to allow edges to be sold in 'bundles', rather than as individual edges.

This type of auction, where agents own multiple edges, has been previously studied (e.g. [10]). However, one fundamental difference in our approach is that they assume the ownership information is private, and agents may not always honestly reveal which edges they own. They then study truthful mechanisms based on this assumption, and show negative results for this setting — that no mechanism which only requires the edge costs to be reported can also be truthful.

That general idea, of studying mechanisms where ownership may be dishonestly reported has also been studied by Kempe et al. in [23]. They consider the concept of mechanisms that are *false-name-proof*. In this setting the auctioneer does not know all of the set-system and the element ownership information. Thus agents can declare this information to the auction, which would allow manipulations such as falsely claiming an element that they own to be a set of agents, or to claim that individual elements are owned by separate agents. A false-name-proof mechanism would not allow any agent to gain greater utility by making a false declaration as to ownership. However, they show a lower bound on frugality of $\Omega(2^n)$ for all false-name-proof mechanisms. These results suggest that allowing agents to dishonestly report their ownership information would prove a major obstacle to finding frugal (truthful) mechanisms in this setting. However, in many real cases, the ownership information would be public knowledge, and not manipulable, and we will restrict our analysis to the cases that either the information is public, or analogously, that it will be honestly revealed due to some external incentive.

In this model, we consider that each agent owns some collection of edges, and is willing to provide all of their edges for some fixed cost. We can see that this may have real applications if we think of this as some 'off-peak' or 'surplus capacity' shortest path problem. Where network owners may have surplus capacity, for instance at certain times of day, then they may be willing to sell access to their entire network for some fixed cost. The fixed cost may represent the overhead of enabling access to the service, or simply what the network considers reasonable so as not to impact on their 'regular' sales.

In many cases, such as public transport or telecommunications, it may well be true that additional network use adds little or nothing to the running cost. It is reasonable to assume that having a single payment for access to the entire network may be quite desirable for the seller — it may be simpler to setup, and more attractive to potential buyers than dealing with smaller network bundles. We can see such an approach used by rail or bus companies in the UK, who will often provide some unlimited off-peak travel opportunities for a single payment [31, 30].

4.2 Problem Definitions and Examples

We now present definitions for the problems that we study here.

4.2.1 Problem Definitions

Name SHORTEST PATH WITH k-SETS (SPk) (unweighted version)

Instance A graph $G = (V, E)$ with two distinct vertices s, t and a collection, $Z = \{Z_1, \ldots, Z_m\}$ of subsets of E ($Z_i \subseteq E$) such that $|Z_i| \leq k$ (each 'bundle' has at most k edges, for some integer parameter k) and $\forall e \in E$, e occurs in exactly one Z_i.

Output The minimum size subset $S \subseteq Z$ that contains a path from s to t. i.e. that there is a path $P \subseteq E$ from s to t in G such that $\forall e \in P; \exists Z_i \in S$ such that $e \in Z_i$.

Name SHORTEST PATH WITH k-SETS (SPk) (weighted version)

Instance A weighted graph $G = (V, E, w)$ with two distinct vertices s, t and a collection, $Z = \{Z_1, \ldots, Z_m\}$ of subsets of E ($Z_i \subseteq E$) such that $|Z_i| \leq k$ (each 'bundle' has at most k edges, for some integer parameter k) and $\forall e \in E$, e occurs in exactly one Z_i. There is a weight function $w(Z_i)$ such that $w(Z_i) \in \mathbb{Q}$ and $w(Z_i) \geq 0$.

Output The subset $S \subseteq Z$ with minimum weight $w(S)$, that contains a path from s to t. i.e. that there is a path $P \subseteq E$ from s to t in G such that $\forall e \in P; \exists Z_i \in S$ such that $e \in Z_i$.

PROBLEM 1: SHORTEST PATH WITH k-SETS

We now define other problems that will be used later on to show hardness results for SPk.

Name EXACT COVER BY 3-SETS (X3C)

Instance A finite set X with $|X| = 3q$ for some $q \in \mathbb{Z}^+$ and a collection, C, of 3-element subsets of X

Output Does C contain an exact cover, i.e. a subcollection $S \subseteq C$ such that every element in X occurs in exactly one member of S?

PROBLEM 2: EXACT COVER BY 3-SETS

X3C is well-known to be NP-complete [14].

Name MINIMUM SET COVER (MSC)

Instance A finite set X and a collection, C, of subsets of X

Output The size of a minimum set cover for X; i.e the minimum size $|S|$ of a sub-collection $S \subseteq C$ such that every element in X occurs in at least one member of S.

PROBLEM 3: MINIMUM SET COVER

MSC is well-known to be NP-hard [14].

Name MINIMUM k-SET COVER (MkC)

Instance A finite set X and a collection, C, of subsets of X, such that $\forall C_i \in C, |C_i| \leq k$.

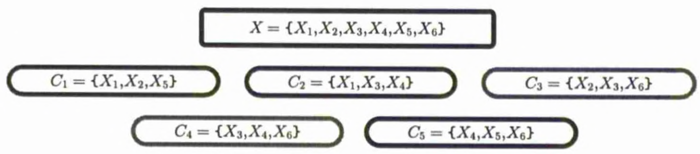
Output The size of a minimum set cover for X; i.e the minimum size $|S|$ of a sub-collection $S \subseteq C$ such that every element in X occurs in at least one member of S.

PROBLEM 4: MINIMUM k-SET COVER

MkC is known to be NP-hard when $k \geq 3$ and in P when $k < 3$ [14]. (Although a hardness proof will be given as a reduction from X3C).

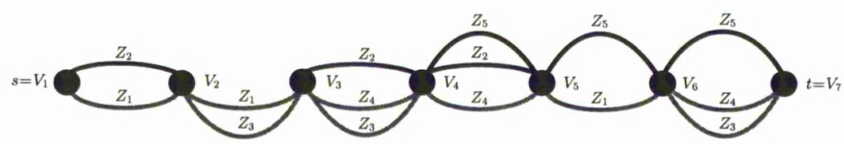
4.2.2 Example

In Figure 4.1 we see an example of an instance of MkC, when $k = 3$ and in Figure 4.2 we see a similar example of an instance of SHORTEST PATH WITH k-SETS, when $k = 3$. The purpose of these two examples is to demonstrate the connection between the two problems, which is that any instance of MkC can easily be transformed into an instance of SPk. This will be shown in more detail later, but it is possible to see from these examples that any solution to Figure 4.1 implies a solution of the same size in Figure 4.2, and vice-versa.



$$S = \{C_1, C_4\}$$

FIGURE 4.1: Example of M3C



$$S = \{Z_1, Z_4\}$$

FIGURE 4.2: Figure 4.1 as SHORTEST PATH WITH 3-SETS

4.3 Hardness results

We can observe that the SPk problem is, when $k = 1$, exactly equivalent to the shortest-path problem that has been well-studied. Furthermore, it is well-known that this problem can be solved optimally in time that is polynomial in the number of edges, e.g. using Dijkstra's algorithm [9]. So, it is an obvious question to ask about the complexity of computing exact solutions when $k \geq 2$, and that is addressed in this section.

For this section, we need only consider the unweighted version of the SPk problem; we will generalize to the weighted version when we examine this problem in an auction setting. Firstly, we will examine the SHORTEST PATH WITH 3-SETS problem, and show that this is NP-hard. This will be achieved by a reduction from MINIMUM 3-SET COVER (M3C), which we will see is a generalization of the known NP-hard problem

EXACT COVER BY 3-SETS (X3C). We then see a reduction from $k = 3$ to $k = 2$, showing that the problem is NP-hard even when each bundle only contains 2 edges, giving a complete characterization for all values of k (namely, in P for $k = 1$ and NP-hard otherwise).

4.3.1 SHORTEST PATH WITH 3-SETS

Proposition 4.1. *M3C is a generalization of the NP-hard problem EXACT COVER BY 3-SETS.*

Proof. Assume that we have an input to EXACT COVER BY 3-SETS that has $|X| = 3q$ for some $q \in \mathbb{Z}^+$. For this input, EXACT COVER BY 3-SETS(X) is equivalent to the decision problem 'Is there a solution to M3C(X) with size $\frac{|X|}{3}$?'. Hence if M3C can be computed exactly then this can be used to give an answer to any instance of the problem EXACT COVER BY 3-SETS, which is well-known to be NP-hard (see, e.g., [14]). □

Lemma 4.2. *There is a polynomial-time reduction from MkC to SPk such that the optimal solution to any instance of MkC can be found with an optimal solution to a given instance of SPk.*

Proof. Taking the input (X, C) from an instance of MkC, when $m = |X|$ and $n = |C|$, build an instance (G, Z)[1] of SPk as follows. Add $m + 1$ new vertices, labelled $V_1, \ldots, V_{m+1}$, and let $s = V_1$ and $t = V_{m+1}$.

For each of the subcollections $C_i \in C$, add a bundle $Z_i \in \mathcal{E}$, and for each element e in C_i give the corresponding bundle Z_i an edge from V_e to V_{e+1}. Hence, every subcollection is mapped to a single agent, and each element in a subcollection is mapped to an edge going from the corresponding vertex to the next vertex in the path. Hence, for every set of subcollections that contains all of the elements of the groundset, there is a corresponding set of bundles that contain all the edges from s to t.

The construction was demonstrated in Figures 4.1 and 4.2. The groundset in Figure 4.1 consists of $\{1, \ldots, 6\}$, hence the set of vertices are $\{1, \ldots, 7\}$ in Figure 4.2. Collection C_1 contains elements $1, 2$ and 5, and so bundle Z_1 has corresponding edges from V_1, V_2 and V_5 (every edge simply connects vertex V_i to V_{i+1}). This is repeated for the other subcollections C_2, C_3, C_4, C_5 in Figure 4.1 to give the edges shown for bundles Z_2, Z_3, Z_4, Z_5 in Figure 4.2

We can observe that the optimal solution in Figure 4.1 is of size 2; subcollection C_1 has elements $\{1, 2, 5\}$ and C_4 has $\{2, 3, 6\}$ and so all 6 elements are covered. The corresponding optimal path in Figure 4.2 is also of size 2 and contains bundle Z_1 with edges from V_1, V_2, V_5 and bundle Z_4 with edges from V_2, V_3, V_6. As there are edges from $V_1, V_2, V_3, V_4, V_5, V_6$ then there is a complete path from $s = V_1$ to $t = V_7$.

[1] G is actually presented as a multigraph for simplicity, but Section 4.4.4 shows that this is not important.

Returning to the proof, suppose that we have an instance of MkC (X, C), and a corresponding instance of an SPk problem (G, Z), generated as described. We will then see that for any solution S to MkC there exists a solution S' to SPk such that $|S| = |S'|$, and that for every solution S' to SPk a solution S to MkC also exists with $|S'| = |S|$.

Taking $S \subseteq C$ as a solution to MkC (X, C) , construct $S' \subseteq Z$ as a solution to SPk (G, Z) by including a bundle Z_i in S' if and only if the corresponding subcollection C_i is in the solution S. Clearly this implies $|S| = |S'|$, so it is sufficient to see that S' is a valid solution.

For S to be a valid solution, every $j \in X$ is present in some $C_i \in S$. As $C_i \in S \Leftrightarrow Z_i \in S'$ then there is a corresponding $Z_i \in S'$ for every $C_i \in S$. As every $j \in X$ must be present in some $C_i \in S$ (or else S is not a valid solution) then every $j \in \{1, \dots, m\}$ is present in some $Z_i \in S'$. As we add an edge from V_j to V_{j+1} for each $j \in C_i \in C$, then there is an edge from every $V_j : j \in \{1, \dots, m\}$ to V_{j+1} — implying a path of m edges from V_1 to V_{m+1}, proving that S' is a valid solution (as $s = V_1$ and $t = V_{m+1}$).

This procedure will be reversed to see that when $S' \subseteq Z$ is a solution to SPk then there exists S, a solution to MkC (X, C) when $|S| = |S'|$. This is simply created in the same way, and includes C_i in S if and only if the corresponding bundle Z_i is in S'.

As before, clearly $|S| = |S'|$, and an edge between every V_j and V_{j+1} must be contained in some $Z_i \in S'$. The corresponding C_i is included in S and hence every element $j \in X$ is covered by some $C_i \in S$.

This shows that every instance of the problem MkC can be solved exactly with an instance of SPk and the proof is complete. □

Lemma 4.3. SHORTEST PATH WITH 3-SETS *is NP-hard.*

Proof. Using Lemma 4.2 any instance of M3C can be polynomial-time reduced to an instance of SP3. As Proposition 4.1 has shown that M3C is NP-hard, this shows that SP3 is NP-hard. □

4.3.2 SHORTEST PATH WITH 2-SETS

We will now see a reduction from SP3 to SP2, showing that SP2 is also NP-hard.

Taking as input an instance of SP3 (G, Z), we will generate a graph $G' = (V', E')$ and a collection Z' as an input (G', Z') to SP2 as follows.

Let $n = |Z|$, and for each $i \in \{1, \dots, n\}$ create 4 new vertices, labelled $V'_i, V'_{i,a}, V'_{i,b}$ and $V'_{i,0}$. Let W be a set containing these vertices.

$$W = \bigcup_{i=1}^{n} \{V'_i, V'_{i,a}, V'_{i,b}, V'_{i,0}\}.$$

Additionally, add copies of all of the vertices in $V \in G$, to complete the vertex set V'.

$$V' = V \cup W.$$

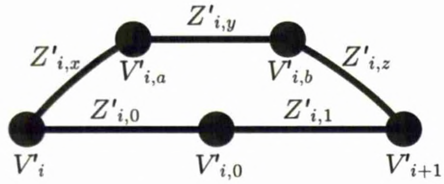


FIGURE 4.3: Construction for V'_i to V'_{i+1}

Let $V'_{n+1} = s$ (combine the last of the new vertices with the vertex marked as s in G).

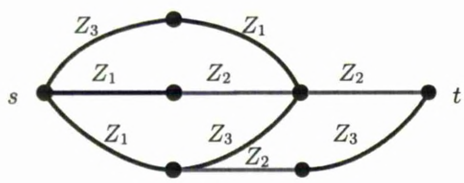
For each $i \in \{1, \ldots, n\}$ define 5 new edges, and let Y be the set of all these edges, as follows.

$$Y = \bigcup_{i=1}^{n} \{(V'_i, V'_{i,a}), (V'_{i,a}, V'_{i,b}), (V'_{i,b}, V'_{i+1}), (V'_i, V'_{i,0}), (V'_{i,0}, V'_{i+1})\}$$

Now copy all of the edges in G, to give the edge set E',

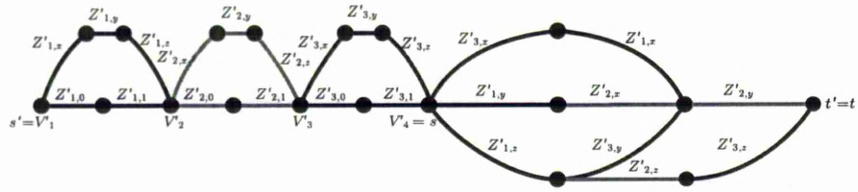
$$\text{let } E' = Y \cup E.$$

Observe that the vertices W and edges in Y give two parallel paths from each V'_i to V'_{i+1}, one of length 3 (via $V'_{i,a}$ and $V'_{i,b}$) and the other of length 2 (via $V'_{i,0}$); we can see this construction in Figure 4.3 as well as the example presented in Figure 4.5.



$$S = \{Z_1, Z_2\}$$

FIGURE 4.4: Example of SHORTEST PATH WITH 3-SETS



$$S = \{Z'_{1,x}, Z'_{1,y}, Z'_{1,z}, Z'_{2,x}, Z'_{2,y}, Z'_{2,z}, Z'_{3,0}, \}$$

FIGURE 4.5: Figure 4.4 as SHORTEST PATH WITH 2-SETS

For each $Z_i \in Z$ if we label the three edges in Z_i as $e_\alpha, e_\beta, e_\gamma$ (the mapping is not important) then define the collections as follows

$$Z'_{i,x} = \{e_\alpha, (V'_i, V'_{i,a})\} \tag{4.1}$$
$$Z'_{i,y} = \{e_\beta, (V'_{i,a}, V'_{i,b})\} \tag{4.2}$$
$$Z'_{i,z} = \{e_\gamma, (V'_{i,b}, V'_{i+1})\} \tag{4.3}$$
$$Z'_{i,0} = \{(V'_i, V'_{i,0})\} \tag{4.4}$$
$$Z'_{i,1} = \{(V'_{i,0}, V'_{i+1})\} \tag{4.5}$$

That is, for every triple Z_i in Z, each edge in Z_i will now be 'paired' (they will both be part of the same 'collection' $Z'_{i,\ldots}$) with any one of the three edges in the three-edge path between V'_i and V'_{i+1}. Both $Z'_{i,0}$ and $Z'_{i,1}$ contain only one edge.

This duplicates every edge in E in E', and recall that $V'_{n+1} = s$, which connects the end of the 'path' construction we created with the vertex s in the copy of the original graph G.

Combine these to give all of the collections

$$Z' = \bigcup_{i=1}^{n} \{Z'_{i,x}, Z'_{i,y}, Z'_{i,z}, Z'_{i,0}, Z'_{i,1}\}.$$

Let the two distinct vertices in G' be $s' = V'_1$, and $t' = t$.

We will now see that any SP3 problem can be solved with this construction of SP2. We will start with two propositions which, when taken together, will show that a minimum solution to either problem will give a minimum solution to the other.

Referring again to the construction in Figure 4.3; informally, we will arrange these so that any $s - t$ path must pass through each of these n constructions in turn. If the upper path is chosen, with three edges, then each of these edges is 'bundled' with one edge from Z'_i, and all three of these bundled edges (belonging to $Z'_{i,x}, Z'_{i,y}, Z'_{i,z}$) may be selected at no additional cost. If the lower path is chosen, with two edges, then no further edges are 'bundled'. Therefore, by choosing the upper path in preference to the lower path, we may select all of Z_i's three edges in G, which 'costs' one edge more than taking the lower path with no additional edges included. We can see that this is equivalent to adding one for every edge Z_i chosen in a solution to SP3(G), with a constant value of two added for every $Z_i \in Z$.

Proposition 4.4. *For any solution $S \subseteq Z$ to* SP3(G) *there is a solution $S' \subseteq Z'$ to* SP2(G') *such that $|S| + 2|Z| = |S'|$.*

Proof. Construct a solution S' from S, as follows. When Z_i is in S include all three of $Z'_{i,x}, Z'_{i,y}$ and $Z'_{i,z}$ in S', otherwise include $Z'_{i,0}, Z'_{i,1}$ in S'

We will see that this is a valid solution in two parts - firstly that there is a path from s' to V'_{n+1}, and secondly a path from V'_{n+1} to t'. We can see that for every $i \in \{1, ..., n\}$ there is a path from V'_i to V'_{i+1} (for each $Z_i \in Z$ then exactly one of the two paths

$Z'_{i,x}, Z'_{i,y}, Z'_{i,z}$ or $Z'_{i,0}, Z'_{i,1}$ is included in S' and either path reaches from V_e to V_{e+1}). Hence there is a path from $s' = V'_1$ to V'_{n+1}.

We will then see that, if Z contains a path from s to t in G then Z' contains a path from s' to t' in G'. For each $Z_i \in S$ then all three of $Z'_{i,x}, Z'_{i,y}, Z'_{i,z}$ are in S' and these contain all of the edges that are in Z_i. This shows that all of the three edges present in each $Z_i \in S$ are contained in some $Z'_{i,x}, Z'_{i,y}$ or $Z'_{i,z}$ that is in S'. If a path were not present in Z' from V'_{n+1} (recall that $V'_{n+1} = s$) to t' in G', then Z could not have contained a path from s to t in G.

As we have seen that there is a path from s' to V'_{n+1} and a path from V'_{n+1} to t' then it follows that there is a path from s' to t'. Now we have seen that S' is a valid solution, we will continue by verifying the size of the solution.

Recalling that $\forall i \in \{1, \dots, n\}$, then S' contains either $Z'_{i,x}, Z'_{i,y}, Z'_{i,z}$ when $Z_i \in S$ or $Z'_{i,0}, Z'_{i,1}$ otherwise. Hence the size of S' can be written as

$$|S'| = \sum_{i=1}^{n} \{3 : Z_i \in S\} + \sum_{i=1}^{n} \{2 : Z_i \notin S\}$$

Which can be rewritten as

$$|S'| = \sum_{i=1}^{n} 2 + \sum_{Z_i \in S} 1$$

and simplified to

$$|S'| = 2n + |S|.$$

Recalling $n = |Z|$, this proves the proposition $|S| + 2|Z| = |S'|$. □

Proposition 4.5. *For any solution $S' \subseteq Z'$ to* SP2(G') *there exists a solution $S \subseteq Z$ to* SP3(G) *such that $|S| + 2|Z| = |S'|$.*

Proof. As an intermediate step, firstly we will create a new, more structured, solution S'' to SP2 having $|S''| = |S'|$. For any $i \in \{1, \dots, n\}$ when S' contains exactly one of $\{Z'_{i,x}, Z'_{i,y}, Z'_{i,z}\}$ and both $Z'_{i,0}$ and $Z'_{i,1}$ then S'' contains all three of $\{Z'_{i,x}, Z'_{i,y}, Z'_{i,z}\}$ but not $Z'_{i,0}$ and $Z'_{i,1}$. For any $i \in \{1, \dots, n\}$ when this does not hold, S'' simply contains whichever of $Z'_{i,x}, Z'_{i,y}, Z'_{i,z}, Z'_{i,0}$ and $Z'_{i,1}$ that were present in S'.

Observe that, as the only edges removed from S' were those contained in $Z'_{i,0}, Z'_{i,1}$ for some i, and these do not contain any edges that are between V_{n+1} and t (i.e. in the G portion of G'), then when S' contains a path from V_{n+1} and t then so does S''.

As the only substitution replaces exactly three edges in S' with three in S'', this tells us that $|S''| = |S'|$. Now construct a solution S, by including Z_i in S if and only if there is at least one of $Z'_{i,x}, Z'_{i,y}, Z'_{i,z}$ in S''. Observe that when $\{Z'_{i,0}, Z'_{i,1}\} \subset S''$ nothing is added to S, otherwise Z_i is added.

Firstly, see that this makes a valid solution - when S'' contains any of $Z'_{i,x}, Z'_{i,y}$ or $Z'_{i,z}$ then S contains Z_i. Suppose, for contradiction, that S does not contain a path from

s to t. As S'' contains a path from s to t', then there must be some edge e_ω (between s and t') that is present in S'' but not in S. (Only the edges between s and t' exist in both G and G'). As e_ω is in S'', it is present in some $Z'_{i,x}, Z'_{i,y}$ or $Z'_{i,z}$ that is in S'' - but not in Z_i. The construction of S includes $Z_i \in S$, whenever $Z'_{i,x}, Z'_{i,y}$ or $Z'_{i,z}$ is in S'' , hence it can only be the case that $e_\omega \notin Z_i$. However, recall that the only edges between s and t' in $Z'_{i,x}, Z'_{i,y}$ and $Z'_{i,z}$ were constructed from the edges in Z_i, giving a contradiction. This shows that when S'' contains a path from s to t', then S contains a path from s to t, making it a feasible solution.

Now, finally, we need to see that $|S| + 2|Z| = |S''|$. We can observe that, if at least two edges from $Z'_{i,x}, Z'_{i,y}$ and $Z'_{i,z}$ were in the lowest-cost solution, S', then both $\{Z'_{i,0}, Z'_{i,1}\}$ would not have been included (as all three of $Z'_{i,x}, Z'_{i,y}$ and $Z'_{i,z}$ would have been a lower-cost path from V'_i to V'_{i+1}). By creating S'', we removed any cases when there is only one edge included from $Z'_{i,x}, Z'_{i,y}$ and $Z'_{i,z}$, hence there are only two possible cases left, either $\{Z'_{i,0}, Z'_{i,1}\} \subset S''$ or $\{Z'_{i,x}, Z'_{i,y}, Z'_{i,z}\} \subset S''$.

Case 1 : $\{Z'_{i,0}, Z'_{i,1}\} \subset S''$

As we have $\{Z'_{i,x}, Z'_{i,y}, Z'_{i,z}\} \cap S'' = \emptyset$ then we can write the contribution to S'' as $|\{Z'_{i,x}, Z'_{i,y}, Z'_{i,z}, Z'_{i,0}, Z'_{i,1}\} \cap S''| = 2$.

Case 2 : $Z'_{i,x}, Z'_{i,y}$ and $Z'_{i,z}$ are all present in S''

Here, $\{Z'_{i,0}, Z'_{i,1}\} \cap S'' = \emptyset$ so we have $|\{Z'_{i,x}, Z'_{i,y}, Z'_{i,z}, Z'_{i,0}, Z'_{i,1}\} \cap S''| = 3$.

So, for all i when $Z_i \in S$ then $|\{Z'_{i,x}, Z'_{i,y}, Z'_{i,z}, Z'_{i,0}, Z'_{i,1}\} \cap S''| = 3$. When $Z_i \notin S$ then $|\{Z'_{i,x}, Z'_{i,y}, Z'_{i,z}, Z'_{i,0}, Z'_{i,1}\} \cap S''| = 2$. This gives us

$$|S''| = \sum_{i=1}^{n} \{3 : Z_i \in S\} + \sum_{i=1}^{n} \{2 : Z_i \notin S\}$$

Which can be rewritten as

$$|S''| = \sum_{i=1}^{n} 2 + \sum_{Z_i \in S} 1$$

and further simplified to

$$|S''| = 2n + |S|$$

Recalling that $n = |Z|$, this gives $|S| + 2|Z| = |S''|$. Recall that $|S'| = |S''|$ and the proof of the proposition is complete. □

Theorem 4.6. SHORTEST PATH WITH 2-SETS *is NP-hard.*

Proof. Taking Proposition 4.4 and Proposition 4.5 together shows that the optimal solutions to SP2 and SP3 differ in size by only the constant additive factor, $2|Z|$.

As SP3 (G, Z, s, t) was already shown to be NP-hard in Lemma 4.3, this shows that SP2 is also NP-hard. It is worth noting that, unlike the reduction from M3C to SP3 in Lemma 4.2, this reduction is not approximation preserving.

□

Corollary 4.7. SHORTEST-PATH WITH k-SETS *is NP-hard for all* $k \geq 2$*,*

Proof. Any instance of SHORTEST-PATH WITH 2-SETS, which was shown to be NP-hard in Theorem 4.6, can be solved by being given as an input to SHORTEST-PATH WITH k-SETS, wherever $k \geq 2$. □

Corollary 4.8. *The weighted version of* SHORTEST-PATH WITH k-SETS *problem is NP-hard for all* $k \geq 2$.

Proof. As the NP-hard unweighted version is simply a special case of the weighted version, (with $w(Z_i) = 1$), the weighted version is also NP-hard. □

4.4 Auction Design

We will now use the weighted version of the SPk (G, Z, w, s, t) problem in a set-system auction setting. Let the set system be $(\mathcal{E}, \mathcal{F})$. Let $\mathcal{E} = \{A_1, \ldots, A_n\}$ be the set of n agents owning the edge bundles $\{Z_1, \ldots, Z_n\}$ respectively. Let $\mathcal{F}$ be the set of solutions for SPk, i.e. $T \in \mathcal{F}$ implies that there is a path, P, between s and t in G, such that $\forall e \in P, \exists A_i \in T$ and $e \in Z_i$ (path P uses only the edges 'owned' by the agents in T). For all $e \in \mathcal{E}$, let $c_e = w(Z_e)$, that is the cost of each agent is equivalent to the weight. As we will be interested in truthful mechanisms, we will also generally assume that $b_e = c_e$.

We can consider that this auction implements the SPk problem; finding a lowest-cost solution for the auction is equivalent to finding a minimum-weight solution to the NP-hard problem SPk.

4.4.1 Using VCG

We saw in Theorem 2.3 that VCG has a frugality ratio of at most $n - 1$. The simpler shortest-path problem (when every edge is owned by a unique agent), is trivially implementable as a special case of SPk. Therefore we could use the example instances from [24] to show a lower bound on frugality of VCG for SPk auctions of $n - 1$. (The same lower bound will be proven later in Lemma 4.12, as for $k = 1$, VCG and the approximation mechanism we use are identical.) However, implementing VCG would require finding an exact solution to the SPk problem. We have seen that finding an exact solution is NP-hard when $k \geq 2$, so we know of no way to do this in polynomial time. As we would like to find some way of running this auction in polynomial time, we will now turn our attention to a polynomial-time mechanism that uses an approximation algorithm.

4.4.2 Mechanism $\mathcal{M}^{\mathcal{P}}$

We will now define a specific mechanism, we will denote this by $\mathcal{M}^{\mathcal{P}}$ This mechanism can be performed in polynomial time, and we will see it has an upper bound on frugality of $k(n - 1)$, which is within a factor of k of the upper bound on frugality for VCG.

Algorithm 2 is an approximation algorithm for SPk. This algorithm computes a shortest path on the underlying graph G, as if each edge was owned separately and given the entire weight (or bid) of the agent that owns it. For each edge e' in the shortest path, where bundle Z_e contains e' then the corresponding agent A_e is included in the winning set.

Algorithm 2: Approximation Algorithm for SPk

1 Take as input an instance of SP$k(G, Z, w, s, t)$;
2 Let $G' = G$. (and hence $V' = V, E' = E, s' = s, t' = t$);
3 Let $w'(e') = w(Z_i)$ for $e \in Z_i$;
4 Compute $S' \subseteq E'$ as the lowest-weight path from s' to t' in G' (use Dijkstra's algorithm);
5 Let $S = \emptyset$;
6 **for each** $e' \in S'$ **do**
7   add A_e to S, when $e' \in Z_e$;
8 return $S^{\mathcal{P}} = S$;

Let $\mathcal{M}^{\mathcal{P}}$ be a mechanism with a selection rule that chooses a solution $S^{\mathcal{P}} \in \mathcal{F}$ by choosing the solution returned by Algorithm 2. As we will be using the bids provided by the agents, in place of weights, let $b_e = w(Z_e)$.

Let the payment rule for this mechanism pay threshold values, and let $p_{\mathcal{E}}^{\mathcal{M}^{\mathcal{P}}}$ be the sum of payments made by mechanism $\mathcal{M}^{\mathcal{P}}$ to all agents in $S^{\mathcal{P}}$.

4.4.3 Frugality Results for $\mathcal{M}^{\mathcal{P}}$

We will aim to show that the mechanism $\mathcal{M}^{\mathcal{P}}$ is truthful, and has a frugality ratio of $k(n-1)$. Firstly, we will see an upper bound on the approximation ratio for the algorithm that is used. Then, once we have confirmed that $\mathcal{M}^{\mathcal{P}}$ is a truthful mechanism, we will use the result about its approximation ratio to determine its frugality.

Proposition 4.9. *Algorithm 2 returns a k-approximation for an exact solution.*

Proof. Let $OPT \subseteq Z$ be an optimal solution to SP$k(G, Z, w, s, t)$. We can construct a path $OPT' \subseteq E'$ in G' using every edge in every agent in OPT. Recall that $\forall e \in \mathcal{E}, |Z_e| \leq k$, hence each agent in OPT contributes at most k edges to OPT' (each with weight $w(Z_e)$ from line 3). The weight of this path can therefore be upper bounded by

$$w'(OPT') \leq w(OPT)k. \tag{4.6}$$

Observe that as OPT is a solution in G, OPT' is a solution in G'. When $S^{\mathcal{P}}$ is a solution in G, we can also consider the path S', (which was chosen in Algorithm 2), which is a solution to the shortest path in G'. As S' contains every edge in $S^{\mathcal{P}}$ then we have

$$w'(S') \geq w(S^{\mathcal{P}}). \tag{4.7}$$

As the selection rule chose S' we know that

$$w'(S') \leq w'(OPT') \tag{4.8}$$

and hence we can arrange inequalities 4.6, 4.7 and 4.8 such that

$$w(S^{\mathcal{P}}) \leq w'(S') \leq w'(OPT') \leq w(OPT)k$$

which simplifies to

$$w(S^{\mathcal{P}}) \leq w(OPT)k.$$

This completes the proof that Algorithm 2 returns a solution that is within a factor of k of the optimal solution, and hence may be called a k-approximation. □

Proposition 4.10. *$\mathcal{M}^{\mathcal{P}}$ is a truthful mechanism.*

Proof. As the mechanism $\mathcal{M}^{\mathcal{P}}$ pays threshold values, it is well known that it is truthful if the selection rule is monotonic (see, e.g., [24]). We will now show that the selection rule in $\mathcal{M}^{\mathcal{P}}$ is monotonic.

Let I and I' be identical instances, other than there is exactly one agent e where $b'_e \neq b_e$.

Assume, for contradiction, that agent e is in the winning set with bid b_e, but is not in the winning set with a lower bid b'_e. Let S be the winning set chosen by $\mathcal{M}^{\mathcal{P}}$ for instance I (with bid vector $\mathbf{b}$) and let T be the winning set chosen by $\mathcal{M}^{\mathcal{P}}$ for I' (with bid vector $\mathbf{b}'$).

We have defined the weights to be equal to the bids, $w(e) = b_e$, so we can consider the bids to the mechanism to be equivalent to the weights that are used in the approximation algorithm.

We know, from the definitions of $\mathbf{b}$ and $\mathbf{b}'$ that

$$b'_S < b_S \text{ (because } e \in S \text{ and } b'_e < b_e) \tag{4.9}$$

$$b_T = b'_T \text{ (because } e \notin T \text{ and all other bids are equal.)} \tag{4.10}$$

As S is chosen as optimal for I, therefore T could not have been a better choice, giving

$$b_S \leq b_T \tag{4.11}$$

similarly, the assumption that T is chosen in I' gives

$$b'_T \leq b'_S. \tag{4.12}$$

By transitivity, from inequalities 4.9 and 4.11 we get

$$b'_S < b_T \tag{4.13}$$

and by substitution of Equation 4.10 into Inequality 4.12 we get

$$b_T \leq b'_S \tag{4.14}$$

then by transitivity of inequalities 4.13 and 4.14 we get

$$b_T \leq b'_S < b_T$$

giving $b_T < b_T$ and hence a contradiction. This shows that given fixed bids by all other agents, no 'winning' agent can become a 'losing' agent by lowering their bid - hence that the selection rule is monotonic and, given the threshold payment rule, the mechanism is truthful.

□

Theorem 4.11. *$\mathcal{M}^{\mathcal{P}}$ is a polynomial time truthful mechanism for* SP*k auctions that has a frugality ratio of at most $k(n-1)$.*

Proof. We use the $\mathcal{M}^{\mathcal{P}}$ mechanism, which is based on the approximation algorithm given in Algorithm 2. We saw in Proposition 4.9 that Algorithm 2 has an approximation ratio of k. Additionally, we saw in the proof of Proposition 4.10 that Algorithm 2 is a monotonic algorithm, and that $\mathcal{M}^{\mathcal{P}}$ is a truthful mechanism. Recall that we saw in Theorem 2.5 that, for all set system auctions, when a mechanism has a monotonic approximation algorithm with an approximation ratio of k, it has a frugality ratio of at most $k(n-1)$, and hence it follows that $\mathcal{M}^{\mathcal{P}}$ has a frugality of at most $k(n-1)$.

□

Lower bounds for $\mathcal{M}^{\mathcal{P}}$

In order to see a lower bound, we will see how to construct an example from any size parameter, $m > k \in \mathbb{Z}$.

Lemma 4.12. *For any $n > 2$, there exists an instance of $\mathcal{M}^{\mathcal{P}}$ that has a payment ratio of $k(n-1)$.*

Proof. The structure can be seen in Figure 4.6, and most importantly consists of a 'long' path from s to t, which has m edges, each owned by a separate agent, which has weight 0. The only alternative solution will consist of a 'short' path of k edges from the same agent, with weight 1. More formally, construct an instance for the size parameter $m \in \mathbb{Z} > k$ as follows.

Let the set of agents be $\mathcal{E} = \{A_0, \dots, A_m\}$ having the edge bundles $\{Z_0, \dots, Z_m\}$ respectively. Define the sets of vertices $V = \{v_0, \dots, v_m\} \cup \{w_0, \dots, w_k\}$. Now define a long path (with m edges) ; $\forall i \in \{1, \dots, m\}$ let $E_i = (w_{i-1}, w_i)$, and define a 'short' path (with k edges) $\forall i \in \{1, \dots, k\}$ let $D_i = (v_{i-1}, v_i)$. Then allocate all edges in the short path to Z_0, let $Z_0 = \{D_1, \dots D_k\}$ and allocate each edge in the long path to one agent; $\forall i \in \{1, \dots, m\}$ let $Z_i = \{E_i\}$.

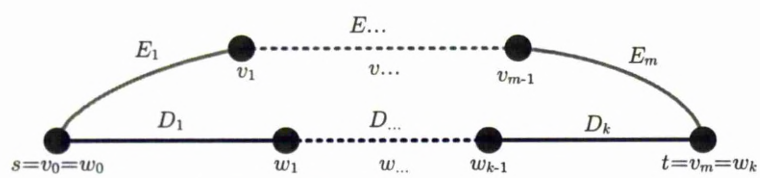
Give Z_0 a weight of 1 (and hence A_0 has a cost of 0, which will be given to the mechanism as a bid for agent A_0) and all other bundles a weight of 0. To define the graph, let $E = \{E_1, \dots, E_k\} \cup \{D_1, \dots, D_m\}$ and let $G = (V, E)$.

Finally, define the start and end points as $s = v_0 = w_0$ and $t = v_m = w_k$.

The lowest-weight path is clearly $\{Z_1, \dots, Z_m\}$, having weight 0. Hence the winning set $S = \{A_1, \dots, A_m\}$ with bids $b_S = 0$. Also we can observe that NTUmin ≤ 1, e.g. $\mathbf{b}^{\min} = (1, 0, \dots, 0)$ satisfies conditions (1),(2) and (3).

The path chosen by the mechanism is the same, $S^{\mathcal{P}} = S$. However, note that if any agent $e > 0$ had a threshold bid $b_e < k$ (with all others being the same) then the mechanism would still choose $S^{\mathcal{P}} = S$ (as $w'(\{Z_0\}) = k$ in G'); therefore the threshold bid for each agent e is $b_e \geq k$, hence $\forall e \in S^{\mathcal{P}}, p_e^{\mathcal{P}} \geq k$. As $|S^{\mathcal{P}}| = m = n - 1$ then we have $p_{\mathcal{E}}^{\mathcal{P}} \geq (n-1)k$ and hence $p_{\mathcal{E}}^{\mathcal{P}} \geq (n-1)k$. The payment ratio is defined as the sum of payments, divided by NTUmin, which is $\frac{p_{\mathcal{E}}^{\mathcal{P}}}{\text{NTUmin}}$. Hence this shows a payment ratio exists of at least $k(n-1)$. □

As Lemma 4.12 shows that, for any $n > 2$, there exists an auction with a payment ratio of $k(n-1)$, which shows that the upper bound on frugality given in Theorem 4.11 is tight.



Bundle	Edges	$w(e)$
Z_0	$\{D_1, \dots, D_k\}$	1
Z_1	$\{E_1\}$	0
$\vdots$	$\vdots$	$\vdots$
Z_m	$\{E_m\}$	0

FIGURE 4.6: Construction of Example for Lower Bound of $\mathcal{M}^{\mathcal{P}}$

4.4.4 Inapproximability Results

Based on the reduction from MkC to SPk (as given in Lemma 4.2) it is clear that the MkC problem can be solved optimally with an instance of SPk. Although the reduction gives an instance of SPk that is presented as a multigraph, suppose that each edge is replaced by the original edge, and an extra edge, in series, which is part of a new subcollection for each extra edge. The resulting simple graph is identical, other than all paths now have twice the length of the original, and hence there is just a constant factor of 2 between the size of the optimal solutions to the original multigraph and the simple graph. Furthermore, the reduction can easily be generalized to include the weighted case for both problems.

It is known that the MkC can be approximated to within a constant factor, but that no better than a constant factor is possible unless P=NP. (see, e.g., [13, 20]). We have already seen in Proposition 4.9 a $k-$approximation for SPk, so it is worth comparing this with the approximation ratios that are known for MkC. The best approximation ratio for MkC remains something of an open problem — the best known result for (weighted) MkC is currently $H_k - \frac{k-1}{8k^9}$, where $H_k = \sum_{1,\ldots k} \frac{1}{k}$ is the $k-$th harmonic number [20].

We may also consider the formulation of SPk which does not provide an upper-bound on the number of edges that are bundled with each agent. Let this version of the problem be called SHORTEST PATH WITH SETS. The reduction given in Lemma 4.2 from MkC to SPk can equally be applied to reduce from MINIMUM SET-COVER to SHORTEST PATH WITH SETS.

Theorem 4.13. *No approximation ratio of $(1-\epsilon)\ln n$ exists for SHORTEST PATH WITH SETS for any $\epsilon > 0$ under the assumption that NP$\subset D_{TIME}(n^{\log\log n})$.*

Proof. Lemma 4.2 has given a reduction which shows that all possible solutions to the MINIMUM SET COVER problem exist with the same weight in an instance of SHORTEST PATH WITH SETS, and vice-versa. Hence, as the solutions have identical weights, the reduction is approximation-preserving.

As we have an approximation-preserving reduction, this shows that any approximation ratio for SHORTEST PATH WITH SETS would give an equivalent approximation ratio for the MINIMUM SET COVER problem. It was shown by Feige in [13], for the MINIMUM SET COVER problem, that no approximation ratio of $(1-\epsilon)\ln n$ exists for any $\epsilon > 0$ under the assumption that $NP \subset D_{TIME}(n^{\log\log n})$. This implies that no approximation ratio of $(1-\epsilon)\ln n$ exists for SPk (with arbitrary k) for any $\epsilon > 0$ under the assumption that $NP \subset D_{TIME}(n^{\log\log n})$. □

Chapter 5

Benchmarks and First-Price Auctions

5.1 Overview

In previous chapters we have considered benchmark values, such as NTUmin and NTUmax, that have previously been seen in the literature for set-system auctions (e.g. [24, 11, 5, 25]). One possible observation is that each can be thought of as an equilibrium in a first-price auction, which was described by Karlin et al. [24]. Recall that, in a first-price auction, the outcome is the price paid when each agent receives exactly their bid value. However, Karlin et al. concentrated on the properties of NTUmin, as a type of equilibrium, but did not suggest any method of finding it (or any other equilibrium). In this chapter we will look at some other possible concepts for types of first-price auction that also reach some sort of equilibrium, and examine the range of payments associated with them. While first-price auctions do not generally incentivize the participants to bid truthfully, they are quite often implemented in reality. As we noted in Chapter 1, some examples are property auctions, which is particularly common practice in Scotland and parts of Australia amongst others [29]. Other items that have been sold in this way include vehicle registration numbers, uranium and radio broadcast licenses.

Before beginning to study other first-price auctions we firstly take a look at some properties of the NTUmin and NTUmax values that we have seen, and consider another method of representing the constraints imposed by conditions (1),(2) and (3) in Definition 1.2. These conditions represent what we might think of as 'fairness' criteria and we will be interested in what we call *feasible* bid vectors, which are those that satisfy these criteria without necessarily being the maximum possible (NTUmax) or the minimum (NTUmin).

Choosing a lowest-cost solution is an obvious goal for the auctioneer, and it is natural that the auctioneer would not wish to pay a large amount for one set when there might be a competing set that would cost less. This notion is captured, more formally, by condition (2) that we saw earlier in Definition 1.2. Looking more closely at condition

(2) tells us that, when comparing the winning set S with some other feasible set T, we can disregard the common part, $S \cap T$, and consider the effect that those outside the winning set (i.e., $T \setminus S$) has on the payments (or bids, in this setting) to the remainder of the winning set ($S \setminus T$).

This tells us that no subset of agents can bid a value which is larger than the bids of some other subset of agents that could constitute a 'replacement' in the winning set. (i.e., as $T \setminus S$ could replace $S \setminus T$ and would make an alternative feasible solution, namely T, we can consider $T \setminus S$ as a possible replacement for $S \setminus T$). Effectively, this means that each feasible set, other than the lowest-cost set S, may provide some upper bound on the bids of some subset of S. Conceptually, it is possibly easier to imagine these alternative sets as providing constraints, and we will see in Section 5.2 how to express the constraints given by condition (2) in Definition 1.2 as a hypergraph.

One of the drawbacks to NTUmin noted by Elkind et al. [11] was that computing NTUmin is NP-hard. However, they note that computing NTUmax does not require condition (3) in Definition 1.2 (as it will be satisfied by maximizing anyway), and hence finding NTUmax is equivalent to solving a linear program. Hence, where a polynomial-time *separation oracle* exists for the constraints of (2) then NTUmax may be computed in polynomial time (e.g. using the well-known *ellipsoid method* [26, 27]). A *separation oracle* is some function that, given a possible solution, can either determine if the solution satisfies the constraints or can indicate a constraint which is not satisfied. A function that finds the lowest-cost feasible set could be used to construct a separation oracle, as follows. Given some possible bid vector, if the feasible set that is returned was equal to S (with tie-breaking in favour of S) then no constraint was violated, and the bid vector is *feasible.* If a different feasible set T was returned, then this would indicate the constraint that was violated ($b_{S \setminus T} \leq c_{T \setminus S}$). Hence, when the lowest-cost feasible set can be found in polynomial time, a polynomial-time separation oracle exists and NTUmax can be computed in polynomial time. Some of the procedures that are described in this chapter also make very similar use of a separation oracle to find the solution to a linear program.

In Section 5.3 we begin our analysis with a procedure inspired by the progressive auction introduced by Demange, Gale and Sotomayor in [8]. In their auction, the auctioneer begins by starting at a low price and repeatedly increases the prices on any items that are *over-demanded* (more than one agent is prepared to pay the current price for that item). We take a similar approach, translated it into a reverse setting. We start with high prices, and repeatedly decrease prices when there is still an 'over-supply', until we reach an equilibrium — when no agent would be willing to decrease their price any further. We do this by choosing a 'current' winning set, then offering the new price to agents not in the winning set. Those who are not currently chosen can either agree to the new price, or their cost value, whichever is higher. Hence, when there is at least one agent that is not currently chosen but can lower her price, we have an 'over-supply' situation and have not yet reached equilibrium.

We show that such an approach can not only give us a complete range of what we consider fair results (between NTUmin and NTUmax), it may also give some that are unfair (in the sense of not satisfying conditions (1),(2) and (3)) and hence we believe that it is not suitable for a reasonable benchmark.

We then consider a special case in Section 5.4 when we allow only simultaneous bid decrements, so that all agents must lower their bids uniformly. We show that this may still produce 'unfair' results.

Alternative processes, in Sections 5.5 and 5.6, begin by choosing an optimal winning set based on the costs, and allowing iterations of bid raises. We show that this does meet the fairness criteria but can result in any of the complete range of values from NTUmin to NTUmax. Even when this is restricted to simultaneous raising, in Section 5.7, we see an almost-complete range of possible values. We do note that, when the problem of finding an optimal feasible set can be solved in polynomial time, then we have a polynomial-time separation oracle and this value may be computed in polynomial time. Additionally, we consider versions where the agents begin from bidding zero and subsequently raising their bids. As much of the literature regarding frugality began with the study of path auctions (e.g. [4, 36, 12]), Section 5.8 considers these different approaches in the special cases of path auctions.

Section 5.9 then modifies this method so that we consider a strictly ordered approach. Every agent is called upon, in turn, to submit a bid value. We assume that each agent will raise its bid as much as possible while remaining in the winning set, hence achieving the fairness criteria, as all agents have an opportunity to raise their bids. We see, via examples, that this procedure may indeed limit the range of values. It depends on the set system as to which part of the range is favoured by this approach: in some cases no value close to NTUmax may be obtained, and in some others, no value close to NTUmin. Given an ordering, and when a polynomial-time separation oracle exists, this may also be computed in polynomial time. However, finding an ordering that gives a minimum value (OMBmin) is not only hard to compute, we show that it is hard to find an order that approximates the minimum value. We can then use instances when NTUmin and OMBmin are equal to show that NTUmin is equally hard to approximate (extending the previous NP-hardness result of [11]).

5.2 Hypergraph Representation of Constraints

Sometimes it can be difficult to see how the feasible bid vectors for a set-system auction are related to each other. In order to assist with visualizing this, we propose a method of representing the constraints, which are implied by a set-system auction, in the form of a hypergraph. While understanding this representation is not critical, it does help to illustrate the underlying structure of some of the examples. Furthermore, in Section 5.9, we will see how the constraints of certain set-system auctions can be represented as a graph, which is an important reduction in the hardness proof given in Section 5.9.6.

Fixing S, we can see that the costs of agents outside S are referred to in conditions (2) and (3), and only take the form of $c_{T\setminus S}$ for some $T \in \mathcal{F}$. Therefore we can, without loss of generality, consider only the sum of costs in each set, rather than the costs of individual agents. Specifically, each feasible set, $T \in \mathcal{F} \setminus \{S\}$, imposes a constraint on the total bid of agents in $S \setminus T$ as follows,

$$b_{S\setminus T} \leq c_{T\setminus S}.$$

It is obvious that for each distinct subset of S there is at most one value of $c_{T\setminus S}$ that provides a minimal (and hence meaningful) constraint in terms of conditions (2) and (3). Hence, if there is some feasible set $T' \in \mathcal{F} \setminus \{S\}$ such that $S \setminus T = S \setminus T'$ and $c_{T\setminus S} < c_{T'\setminus S}$, then the constraint implied by T' is not minimal, and we do not need to include T' in a representation of the constraints.

In order to try and make the structure easier to visualize, we will now represent these restrictions in the form of a hypergraph $H = (X, E)$ with a weight function w on the hyperedges. Make each agent e in S a vertex in the hypergraph

$$\text{Let } X = \{e_1, \ldots, e_m\}.$$

For each feasible set, T, that 'minimally constrains' some subset of S, add a hyperedge, E_e, to exactly those vertices

$$\forall T \in \mathcal{F}, \text{let } E_e = S \setminus T$$

and weight the hyperedges with the value of the constraint.

$$\forall T \in \mathcal{F}, \text{let } w(E_e) = c_{T\setminus S}.$$

Example 5.1 shows how these constraint sets and values are constructed from a set-system auction.

Example 5.1.

Given 8 agents $\{1, \ldots, 8\}$ with costs given in Table 5.1.

Suppose we have feasible sets $\mathcal{F} = \{T_1 = \{A_1, A_2, A_3, A_4, A_5, A_6\}, T_2 = \{A_7, A_3, A_5, A_6\}, T_3 = \{A_7, A_2, A_4, A_6\}, T_4 = \{A_8, A_1, A_4, A_5\}\}$;

Thus $S = T_1$ is the cheapest feasible set.

Agent	c_e	
A_1	0	
A_2	0	
A_3	0	
A_4	0	
A_5	0	
A_6	0	
A_7	1	
A_8	2	
Constraint set $(S \setminus T_i)$	Agents	Constraint value $(\bar{c}_{T_i \setminus S})$
$S \setminus T_2$	$\{1, 2, 4\}$	1
$S \setminus T_3$	$\{1, 3, 5\}$	1
$S \setminus T_4$	$\{2, 3, 6\}$	2

TABLE 5.1: Showing Constraint Sets for a Set-System

Table 5.1 shows constraint set and the corresponding constraint values. The feasible set with lowest cost is T_1, with cost 0. Hence all the agents in T_1 are the vertices of the hypergraph (they are $A_1, \dots, A_6$). Next, take feasible set T_2, and let $(S \setminus T_2)$ be the constraint set. Condition (2) in Definition 1.2 gives us $b_{S \setminus T_2} \leq c_{T_2 \setminus S}$ which can be rewritten as $b_{\{A_1, A_2, A_4\}} \leq c_{\{7\}}$. Hence $c_{\{7\}} = 1$ is a constraint value for the constraint set $S \setminus T_2$, and $b_{S \setminus T_2} \leq 1$, which is indicated on the hypergraph by the hyperedge containing $\{A_1, A_2, A_4\}$ with weight 1. The same process can be applied to the two other feasible sets, T_3 giving $b_{\{A_1, A_3, A_5\}} \leq 1$, and T_4 giving $b_{\{A_2, A_3, A_6\}} \leq 2$, both as shown in the hypergraph in Figure 5.1.

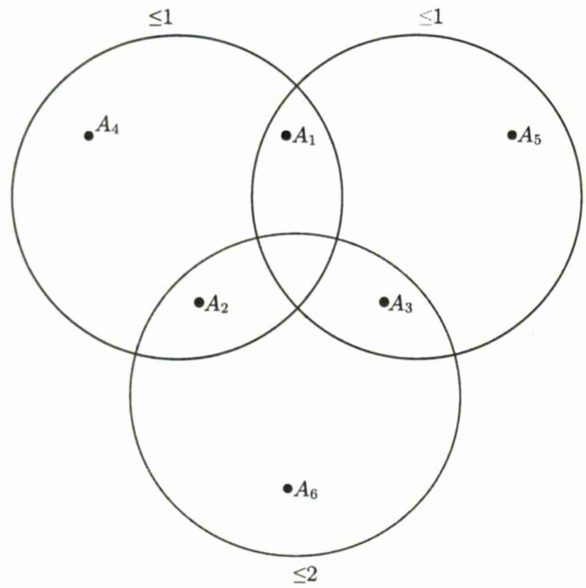


FIGURE 5.1: Hypergraph Representation of Constraints for Table 5.1

5.3 Descending Price Auction

In this section we consider procedures similar to those of Demange et al. [8]. Recall that, in their auction, the auctioneer begins by selling items at a low price, and increases prices on any items that are over-demanded — and hence due to the competition for those items, the price can be increased. We firstly take a similar approach, but translate it into a reverse setting. We start off by buying at high prices, and then decrease prices when there is still an 'over-supply', until we reach an equilibrium — where no agent would be willing to decrease their prices any further.

We analyse this class of descending-price auctions and find that while any feasible bid vector can result, the scheme is not guaranteed to result in a feasible bid vector.

Algorithm 3 gives a definition for such an auction, for a given function f and let β be the bid vector returned by this Descending Price Auction. The function f uses the current state of the auction (i.e., the instance and a current vector of bids) to decide which agent will be called upon next to reduce its bid. The agent chosen by f will then be asked to reduce its bid, and the process will repeat until no agent would be willing to reduce their bid any further.

A more intuitive description is that the procedure initially sets the bids to large values, identifies a winning set S and then repeatedly allows those outside the winning set, in some order, to lower their bid values by some small amount ϵ with the aim of entering the winning set. It is assumed that a bidder whose cost has been reached, would decline to reduce his bid and would exit the competition. We repeat this 'choosing and lowering' process until all losing agents bid their cost; hence no further reductions can take place and the process terminates. (For the sake of consistency, we call the final winning set S; observe that this will be equal to a lowest-cost feasible set assuming a suitable choice of ϵ and tie-breaking rules). As we will see, the choice of the selection function, f, allows the auctioneer to create a wide variety of outcomes to this auction process.

We could consider a slightly more complex version of this auction, whereby an agent is chosen and simultaneously given a price that it must reduce its bid to (if possible). It is easy to observe that, given a small enough parameter ϵ, this could be simulated by simply calling upon the same agent repeatedly until the desired price is reached. For that reason, and the additional simplification of the selection function, here we consider just the version with an ϵ parameter and assume that all bids and costs are an exact multiple of ϵ (such as for a discrete currency value).

Algorithm 3: A Class of Descending Price Auctions

The algorithm defines a class of auctions, each is uniquely defined by the selection function $f(I, \mathbf{b}')$. The set T represents the current winning solution, given bids $\mathbf{b}'$.

Let ϵ be some small value, as a parameter of the auction;
Let $t \in \mathbb{R}$ be some large value such that $t > c_{\mathcal{E}} n$;
for each $e \in \mathcal{E}$ **do**
    Let $b'_e = t$;
Let $T \in \operatorname{argmin}_{R \in \mathcal{F}} \mathbf{b}'_R$;
When at least one agent, not currently chosen in T, can still reduce its bid then the selection function f will choose one such agent to reduce its bid by ϵ.
while $\exists i \notin T$, *such that* $b'_i > c_i$ **do**
    Let $e = f(I, \mathbf{b}')$;
    Let $b'_e = \max(b_e - \epsilon, c_e)$;
    Let $T = \operatorname{argmin}_{R \in \mathcal{F}} \mathbf{b}'(R)$;
Let $S = T$;
Let $\beta = \mathbf{b}'$;
return S, β;

Example 5.2 shows that we may get a bid vector as low as c_S, even when $c_S <$ TUmin $<$ NTUmax.

Example 5.2. *(a commodity auction:) Suppose there are* 4 *agents, each $A_e \in \{A_1, A_2, A_3, A_4\}$ with a quantity q_e and cost c_e given in Table 5.2. Assume the buyer desires* 3 *units.*

	Agent	q_e	c_e	$b_e^{\min}$	$b_e^{\Downarrow,f}$
	A_1	2	0	1	0
	A_2	1	0	0	0
	A_3	1	0		
	A_4	3	1		
Total				1	0

TABLE 5.2: Descending Price Auction Reaches 0

We will describe how the final bid vector β may be reached as a three-stage process, as follows.

Stage 1: When $b_4' > 1$.

When $\{A_4\}$ is the winning set, call upon A_2 to reduce its bid when $b_2' > b_4'$, or call upon A_1 otherwise. When $\{A_4\}$ is not the winning set, call upon A_4 to reduce its bid. This stage can be repeated whenever $b_4' > 1$, as A_4 can always be called upon unless $T = \{A_4\}$, and then we can call upon A_1 or A_2 (as above) until $b_1' + b_2' \leq b_4'$ at which point $T = \{A_1, A_2\}$ becomes the lowest-cost winning set (due to lexicographical ordering).

Stage 2: When $b_4' = 1$ and $b_1' > 0$.

Call upon A_1 to reduce its bid. Because A_3 has not yet reduced its bid, and we only called upon A_2 when $b_2' > b_4'$ then we have $b_3' \geq b_2' \geq b_4'$. Hence the current winning set is $T = \{A_4\}$ and A_1 can be repeatedly call upon to reduce its bid until $b_1' = 0$.

Stage 3: When $b_4' = 1$ and $b_1' = 0$.

Call upon A_2 or A_3 to reduce its bid, whichever is not in the current winning set T. When $b_3' = 0$ then A_2 can be called upon to reduce its bid until $b_2' = 0$ and the winning set is given by $T = \{A_1, A_2\}$ due to lexicographical ordering.

While, in this example we have $c_S = 0$, it is trivial from the algorithm that c_S is a lower bound, as each bid chosen must be at least the cost of the agent (in line 8 of Algorithm 3).

One feature of this auction is that, by choosing the method of selecting agents, the auctioneer may be able to obtain *any* feasible bid vector as a result. A proof of this follows.

Theorem 5.1. *Any feasible bid vector,* **b** *(that satisfies conditions (1),(2) and (3)) is achievable with a Descending Price Auction.*

Proof. The definition of the Descending Price Auction begins with a bid vector $\mathbf{b}'$, such that $\forall e \in \mathcal{E}, b'_e > c_{\mathcal{E}}n$. Observe that NTUmax $\leq c_{\mathcal{E}}n$, and that for any feasible bid vector $\mathbf{b}$ then $\forall e \in S, b_e \leq$ NTUmax by definition of NTUmax, and hence $\forall e \in S, b_e \leq$ NTUmax $\leq c_{\mathcal{E}}n$.

This shows that the starting state meets the condition $\forall e \in \mathcal{E}, b'_e > b_e$.

We now claim that, providing this condition $\forall e \in \mathcal{E}, b'_e > b_e$ is maintained, then Algorithm 3 must reach $\mathbf{b}' = \mathbf{b}$ (and terminate with this), as follows.

Let $\mathbf{b}$ be any feasible bid vector and let β be the bid vector obtained from the Descending Price Auction. We will consider that $\mathbf{b}$ is a 'target' vector — that is the bid vector that the auctioneer wishes to obtain. We can now see that as long as f chooses e such that $\forall e \in \mathcal{E}, b'_e \geq b_e$ and $\forall e \notin S, b'_e \geq c_e$ then we will reach $\beta = \mathbf{b}' = \mathbf{b}$.

We claim that when $\mathbf{b}$ is a specified feasible bid vector, $\mathbf{b}'$ is a current bid vector in Algorithm 3 and the following hold; $\forall e \in S, b'_e \geq b_e$ and $\forall e \notin S, b'_e \geq c_e$, then there exists a function f, such that $\mathbf{b}' = \mathbf{b}$ is achievable with a Descending Price Auction as follows.

Let S be the lexicographically first lowest-cost feasible set (which will be the set returned by Algorithm 3).

If we can show that, unless $b'_S \leq b_S$, there is always some agent e that can be chosen by f to reduce its bid, then reaching $\mathbf{b}' = \mathbf{b}$ is inevitable, as trivially when both $b'_S \leq b_S$ and $\forall e \in S, b'_e \geq b_e$ hold then $\mathbf{b}' = \mathbf{b}$. We do this as follows;

Let T be the current winning set, that has been chosen by Algorithm 3 at Line 9.

Case 1: $S \neq T$.

If T is chosen in preference to S, then we must have $b'_T < b'_S$ (as ties are broken lexicographically, both when selecting S and T). This trivially implies that

$$b'_{T\setminus S} < b'_{S\setminus T} \tag{5.1}$$

(as each can be added to $S \cap T$ to make a feasible set). As we know that the algorithm does not allow bids below the cost, then we must have

$$c_{T\setminus S} \leq b'_{T\setminus S}.$$

By transitivity (with Inequality 5.1), this gives us

$$c_{T\setminus S} < b'_{S\setminus T}.$$

We know, from the fact that S is a cheapest solution, that $c_{S\setminus T} \leq c_{T\setminus S}$ (or else we could add $S \cap T$ to each and we would have $c_T < c_S$). Hence, by substitution, we have

$$c_{S\setminus T} < b'_{S\setminus T}.$$

Therefore there is some agent $e \in (S \setminus T)$ such that $c_e < b'_e$ and hence there exists some agent $e \notin T$ that can be chosen by f to reduce its bid.

Case 2: $S = T$.

When $S = T$ and $b'_S > b_S$ (from the property we wish to maintain) then there is some e such that $b'_e > b_e$. As $\mathbf{b}$ is a feasible bid vector, then it must satisfy (3), and hence for e there is some T_e such that $b_{S\setminus T_e} = c_{T_e\setminus S}$ (and $e \in S \setminus T_e$). We have specified that there is no j such that $b'_j < b_j$, and we have $b'_e > b_e$, then it follows that

$$b'_{S\setminus T_e} > b_{S\setminus T_e}. \tag{5.2}$$

In order for the algorithm to choose S, instead of T_e, then it must not have a higher bid — hence $b'_{S\setminus T_e} \leq b'_{T_e\setminus S}$ (this is trivially implied by $b'_S \leq b'_{T_e}$). By including Inequality 5.2 we can rewrite this as

$$b_{S\setminus T_e} < b'_{S\setminus T_e} \leq b'_{T_e\setminus S}$$

and substituting the equation from condition (3) in Definition 1.2 again ($b_{S\setminus T_e} = c_{T_e\setminus S}$) we get

$$c_{T_e\setminus S} < b'_{T_e\setminus S}$$

therefore there is some agent $j \in T_e \setminus S$ (and hence not in $S = T$) that can be selected to reduce its bid next.

This has shown that, providing that $b'_S > b_S$, function f can always choose some agent to reduce its bid. By repeatedly choosing such an agent, we must eventually reach $b'_S \leq b_S$, and hence that any 'target' vector that is feasible can be achieved with a Descending Price Auction. Clearly, condition (2) in Definition 1.2 implies that once we have reached such a target vector, and all other bids are at least their cost, then there is no other feasible set that would be chosen in preference to S, and the algorithm will terminate, with $\mathbf{b}' = \mathbf{b}$. Therefore, there is a selection function f, for which Algorithm 3 will return $\beta = \mathbf{b}$ for any feasible bid vector $\mathbf{b}$.

□

5.4 Uniformly Descending Price Auction

A natural-looking restriction of the Descending Price Auction is to give the agents initial bids equal to some large (common) value, and then call upon them to reduce the bids in a round-robin manner. Thus, the prices of bidders who continue to stay in the competition would go down at the same rate.

A formal definition is given in Algorithm 4.

If we consider this with the simple commodity auction in Example 5.3 it shows that we may not always get a feasible bid vector from this process.

Algorithm 4: A Uniformly Descending Price Auction

The current bid value is represented by t, and starts at a large value $t > c_{\mathcal{E}}n$. All agents that are not in the current winning set S will reduce their bids, as t reduces with each iteration until either they are chosen in S, or their bid reaches their cost value.

Let $g(x, \mathbf{a}, X) = \mathrm{argmin}_{T \in \mathcal{F}} b_T$ where $\forall e \in X, b_e = a_e$ and $\forall e \notin X, b_e = \max(x, c_e)$;

A function $g(x, \mathbf{a}, X)$ which returns the lexicographically first, lowest-cost feasible set, based on values from the input vector (if the agent is in the current winning set specified), or the larger of the input value given and the agent's cost (for those agents not in the current winning set – they would like to lower their bids to the specified value, if possible).

Let $t \in \mathbb{R}$ be some large value such that $t > c_{\mathcal{E}}n$;

Let $S = \emptyset$;

Let $\mathbf{b}' = ()$;

while $\exists t' < t$ *such that* $g(t', \mathbf{b}', S) \neq S$ **do**

  maximize t' **subject to**

    $g(t', \mathbf{b}', S) \neq S$;

    $t' < t$;

    Finds the next interesting value of t, which is when the winning set can change.

  Let $t = t'$;

  Let $S = g(t, \mathbf{b}', S)$;

  for each $e \notin S$ **do**

    $b'_e = \max(t, c_e)$;

Let $\mathbf{b}^{\downarrow} = \mathbf{b}'$;

return $S, \mathbf{b}^{\downarrow}$;

Example 5.3. *(a commodity auction:) Suppose there are 4 agents, each $A_e \in \{A_1, A_2, A_3, A_4\}$ with a quantity q_e given in Table 5.3. Assume the buyer desires 3 units.*

	Agent	q_e	c_e	b_e^{TUmin}	$b_e^{\downarrow}$
	A_1	2	0	1	1/2
	A_2	1	0	0	0
	A_3	1	0		
	A_4	3	1		
Total				1	1/2

TABLE 5.3: Descending Price Auction as low as TUmin/2

Observe that there are three (minimal) feasible sets, they are

$$\mathcal{F} = \{\{A_1, A_2\}, \{A_1, A_3\}, \{A_4\}\}.$$

We can verify the descending bid values given in Table 5.3 by following the process of Algorithm 4 on this instance as follows. When $t = 1$ *at line 9, we must have* $S = \{A_4\}$, *as if* $S \neq \{A_4\}$ *then* $b'_4 = 1$ *and therefore* $b'_4 < b'_S$ *and* $\{A_4\}$ *is chosen. Then, when we reach* $t = 1/2$ *we will have either* $S = \{A_1, A_2\}$ *or* $S = \{A_1, A_3\}$ *and these two feasible sets may alternate until* $t = 0$, *giving* $b'_1 = 1/2$ *and* $b'_2 = b'_3 = 0$.

5.5 Ascending Price Auction

We will now consider a similar process where the bids ascend (initially from their cost), rather than descend. Again, we will see that by choosing the order in which agents are called upon to raise their bids, the auctioneer is able to obtain any desired feasible bid vector.

Algorithm 5 gives a definition for such an auction, which we will call an Ascending Price Auction.

The agents all start by being given a bid value equal to their cost. A lowest-cost set S is chosen. Then the auctioneer can repeatedly ask some agent $e \in S$ to increase its bid by some small value ϵ. The auctioneer chooses the next agent e by means of some function $e = f(I, \mathbf{b}')$, when I is the instance and $\mathbf{b}'$ is the current bid vector chosen in Algorithm 5. We assume that an agent that is currently chosen will behave rationally and never raise its bid when requested if that would remove it from the winning set. When there are no agents (in S) that can increase their bids and remain in the winning set, then $f(I, \mathbf{b}')$ will return an empty set, and the process will terminate. As we will see, the choice of the selection function, f, allows the auctioneer to create a wide variety of outcomes to this auction process.

Let $\mathbf{b}^{\Uparrow,f}$ be the bid vector obtained by the Ascending Price Auction given selection function f.

Algorithm 5: A Class of Ascending Price Auctions

The algorithm defines a class of auctions, each is uniquely defined by the selection function $f(I, \mathbf{b}')$. *The set* S *represents the winning solution.*

Given some selection function $f(I, \mathbf{b}')$;
Let ϵ by some small value, as a parameter of the auction;
for each $e \in \mathcal{E}$ **do**
    Let $b'_e = c_e$;
Let $S = \operatorname{argmin}_{R \in \mathcal{F}} b'_R$;
Breaking ties lexicographically
while $f(I, \mathbf{b}')$ *is non-empty* **do**
    Let $e = f(I, \mathbf{b}')$;
    Let $b'_e = b'_e + \epsilon$;
Let $\mathbf{b}^{\Uparrow,f} = \mathbf{b}'$;
return $S, \mathbf{b}^{\Uparrow,f}$;

Let $\mathbf{b}$ be any feasible bid vector, which we may consider as a target vector that we wish to obtain. Let $\mathbf{b}'$ be the bid vector obtained during the Ascending Price Auction. We will now verify that as long as function f chooses e in order to maintain the property $\forall e \in S, b'_e \leq b_e$ then the algorithm will reach $\mathbf{b}^{\Uparrow,f} = \mathbf{b}' = \mathbf{b}$.

Theorem 5.2. *Any feasible bid vector* $\mathbf{b}$ *(that satisfies conditions (1),(2) and (3)) is achievable with an Ascending Price Auction.*

Proof. The definition of the Ascending Price Auction begins with a bid vector $\mathbf{b}'$, such that $\mathbf{b}' = \mathbf{c}$. As any feasible bid vector $\mathbf{b}$ must satisfy condition (1) in Definition 1.2, it has $\forall e \in S, b_e \geq c_e$ and we can substitute to give $\forall e \in S, b_e \geq b'_e$.

We will now see that, as long as this property ($\forall e \in S, b_e \geq b'_e$) is maintained during the run of Algorithm 5 then the algorithm will reach $\mathbf{b}' = \mathbf{b}$, and subsequently terminate.

Let S be the lexicographically first, lowest-cost feasible set which is chosen by Algorithm 5.

When $b'_S = b_S$ and $\forall e \notin S, b'_e \leq b_e$, then $\mathbf{b}' = \mathbf{b}$ and hence $\mathbf{b}'$ additionally satisfies condition (2) and condition (3) in Definition 1.2. Therefore, when $\mathbf{b}' = \mathbf{b}$, no agent can raise its bid, and the algorithm will terminate. We have assumed a discrete parameter for ϵ such that no agent e can have consecutive bids, b'_e, b_e, such that $b'_e < b_e$ and $b''_e > b_e$ (i.e., the bids and target are exactly some multiple of ϵ, hence it is not possible to raise to strictly more than its target in one increment). Therefore, we only need to show that, unless $b'_S \geq b_S$, there is always some agent that can be called upon by function f to increase its bid; i.e., that there is some e such that $b'_e < b_e$.

When $b'_S < b_S$ there is some e such that $b'_e < b_e$ (as we may not have a set $V \subseteq S$ such that $b'_V > b_V$).

For every $T \in \mathcal{F}$, condition (2) in Definition 1.2 gives us

$$b_{S \setminus T} \leq c_{T \setminus S}$$

and for every $T_e \in \mathcal{F}$ when $e \in S \setminus T_e$ then (as $b'_e < b_e$) we have

$$b'_{S \setminus T_e} < b_{S \setminus T_e}$$

and from condition (2) in Definition 1.2

$$b'_{S \setminus T_e} < b_{S \setminus T_e} \leq c_{T_e \setminus S}.$$

Therefore, as $b'_{S \setminus T_e} < c_{T_e \setminus S}$ is a strict inequality then, for agent e, there is no T_e set that gives an equation satisfying condition (3) in Definition 1.2, and hence agent e can be chosen to increase its bid next.

This has shown that, providing that $b'_S < b_S$, then there exists some agent e to increase its bid next, which may be chosen by the selection function f. By repeatedly choosing such an agent, we must eventually reach $b'_S \geq b_S$, and hence that any 'target' vector satsfying condtitions (1*),(2), and (3) can be achieved with an Ascending Price

Auction. As described earlier, once we have reached such a vector, then no agent can raise its bid and still be chosen (due to condition (3) in Definition 1.2), and the algorithm will terminate, with $\mathbf{b}^{\Uparrow,f} = \mathbf{b}' = \mathbf{b}$.

This shows that the bid vector $\mathbf{b}$ is achievable with an Ascending Price Auction (and hence for any NTUmin $\leq b_S \leq$ NTUmax). □

5.6 Ascending from Zero Auction

We now see a very similar auction, which differs only in that agents start their bidding from zero, rather than their cost value. The definitions and proofs are otherwise almost entirely duplicated, in an attempt to maintain clarity at the expense of verbosity.

Algorithm 6 gives a definition for this auction.

Algorithm 6: A Class of Ascending from Zero Auctions

The algorithm defines a class of auctions, each is uniquely defined by the selection function $f(I, \mathbf{b}')$. The set S represents the winning solution.

`Given some selection function` $f(I, \mathbf{b}')$;
`Let` ϵ `by some small value, as a parameter of the auction;`
for each $e \in \mathcal{E}$ **do**
    `Let` $b'_e = 0$;
`Let` $S = \text{argmin}_{R \in \mathcal{F}} b'_R$;
Breaking ties lexicographically
while $f(I, \mathbf{b}')$ *is non-empty* **do**
    `Let` $e = f(I, \mathbf{b}')$;
    `Let` $b'_e = b'_e + \epsilon$;
`Let` $\mathbf{b}^{\Uparrow *,f} = \mathbf{b}'$;
return $S, \mathbf{b}^{\Uparrow *,f}$;

Let $\mathbf{b}$ be any bid vector satisfying conditions (1*),(2) and (3) from Definition 1.3, which we may consider as a target vector that we wish to obtain. Recall that condition (1*) is a relaxed version of condition (1) in Definition 1.2, that allows transferable utility, and states that $b_e \geq 0$. Let $\mathbf{b}'$ be the bid vector obtained during the Ascending from Zero Auction, using function f. We will now verify that as long as function f chooses e in order to maintain the property $\forall e \in S, b'_e \leq b_e$ then the algorithm will reach $\mathbf{b}' = \mathbf{b} = \mathbf{b}^{\Uparrow,f}$.

Theorem 5.3. *Any target bid vector* $\mathbf{b}$ *(that satisfies conditions (1*),(2) and (3)) is achievable with an Ascending from Zero Auction.*

Proof. The definition of the Ascending from Zero Auction begins with a bid vector $\mathbf{b}'$, such that $\forall e \in \mathcal{E}, b'_e = 0$ — therefore trivially, this gives $\forall e \in S, b_e \geq b'_e$.

We will now see that, as long as this property ($\forall e \in S, b_e \geq b'_e$) is maintained during the run of Algorithm 6 then the algorithm will reach $\mathbf{b}' = \mathbf{b}$, and subsequently terminate.

Let S be the lexicographically first, lowest-cost feasible set which is chosen by Algorithm 6.

When $b'_S = b_S$ and $\forall e \notin S, b'_e \leq b_e$, then $\mathbf{b}' = \mathbf{b}$ and hence $\mathbf{b}'$ satisfies condition (2) and (3) of Definition 1.2. Therefore, when $\mathbf{b}' = \mathbf{b}$, no agent can raise its bid, and the algorithm will terminate. Again, we only need to show that, unless $b'_S \geq b_S$, there is always some agent that can be called upon by function f to increase its bid; more formally, there is some e such that $b'_e < b_e$.

When $b'_S < b_S$ then there is some e such that $b'_e < b_e$.

For every $T \in \mathcal{F}$, condition (2) in Definition 1.2 gives us

$$b_{S\setminus T} \leq c_{T\setminus S}$$

and for every $T_e \in \mathcal{F}$ when $e \in S \setminus T_e$ then (as $b'_e < b_e$) we have

$$b'_{S\setminus T_e} < b_{S\setminus T_e}$$

and due to condition (2) in Definition 1.2

$$b'_{S\setminus T_e} < b_{S\setminus T_e} \leq c_{T_e\setminus S}.$$

Therefore, as $b'_{S\setminus T_e} < c_{T_e\setminus S}$ is a strict inequality then, for agent e, there is no T_e set with an equation satisfying condition (3) in Definition 1.2, and hence agent e can be chose to increase its bid next.

This has shown that, providing that $b'_S < b_S$, then there exists some agent e to increase its bid next, which may be chosen by some selection function f. By repeatedly choosing such an agent, we must eventually reach $b'_S \geq b_S$, and hence that any 'target' vector that is feasible can be achieved with an Ascending from Zero Auction. As described earlier, once we have reached such a feasible vector, then no agent can raise its bid and still be chosen (due to condition (3) in Definition 1.2), and the algorithm will terminate, with $\mathbf{b}^{\Uparrow,f} = \mathbf{b}' = \mathbf{b}$.

This shows that the feasible bid vector $\mathbf{b}$ is achievable with an Ascending from Zero Auction (that is, for any TUmin $\leq b_S \leq$ TUmax). □

5.7 Uniformly Ascending Auctions

We now consider the special-case of this auction when all of the bids rise at the same, uniform, rate. In this setting it is unnecessary to specify a discrete parameter ϵ as we can compute the exact value that agents may maximally raise their bids to while still remaining in the winning set.

Algorithm 7 gives a definition for this first-price auction, when all bids increase at a uniform rate, until an agent stops bidding any higher because then it would no longer be in the winning set. We call this a 'Uniformly Ascending Auction'. Observe that line 6 ensures that all rises are performed uniformly (on any agents that have not already reached a maximum bid).

Algorithm 7: A Uniformly Ascending Price Auction

Choose S as a lowest-cost solution, then we use a variable t to determine how much more than its cost an agent can bid. Once an agent, e, can bid no higher, e is added to set X (to record this information), and the current bid value b'_e is memorized in b_e. Therefore, when all agents have increased, and none can bid any higher, $X = S$ (where S is the winning set) and $\mathbf{b}$ represents the final bids of all agents $\mathbf{b} = (b_1, \ldots, b_n)$.

1 Let $S = \{1, \ldots, m\} = \text{argmin}_{T \in \mathcal{F}} c_T$;
2 Let $X = \emptyset$;
3 **for each** $e \in \mathcal{E}$ **do**
    Set a starting bid b'_e for each agent $e \in S$.
4   Let $b'_e = 0$;
5 **while** $X \neq S$ **do**
6   **maximize** t **subject to**
        Find the next interesting value of t — the amount that all of the remaining agents (not in X) could simultaneously increase their bids by without changing the winning set.
7   for all $T \in \mathcal{F}, b_{(S \setminus T) \cup X} + c_{S \setminus (T \cup X)} + t|S \setminus (T \cup X)| \leq c_{T \setminus S}$
    Note that this can be solved in polynomial time as an LP where a separation oracle exists, as described in Section 5.1.
8   **for each** $e \in S$ **do**
        Set a current bid b'_e for each agent $e \in S$ — its stored bid b_e if it has already stopped, or else it keeps ascending with t.
9   Let $b'_e = b_e + t$;
10   Let $S' = \text{argmin}_{T \in \mathcal{F}} b'_T$;
11   **for each** $e \in S$ **do**
        If agent e can raise its bid to b'_e and still be chosen in the winning set, then it will now bid b'_e. If this raise would cause it to drop out of the winning set, then it will not raise its bid now, and will stop raising it in future (its 'final' bid value was stored in b_e in a previous iteration.
12   **if** $e \in S'$ **then**
13   Let $b_e = b'_e$;
14   **else**
15   Let $X = X \cup \{e\}$;
16 **return** $S, \mathbf{b}$;

We can observe that this uniformly rising process gives a feasible bid vector, and hence is in the range NTUmin($\mathbf{c}$) to NTUmax($\mathbf{c}$), as follows. As each agent starts from its cost, condition (1) in Definition 1.2 is satisfied, no agent will continue raising when it would no longer be in the winning set, hence condition (2) is satisfied, and each agent continues raising until it must stop, satisfying condition (3).

Also observe that this can be solved with at most n iterations of solving the linear program at line 6, (to determine the next value of t that will result in a change to the current winning set S') so it can be solved in polynomial time wherever there is a polynomial time separation oracle for the constraints. (Recall from Section 5.1 that, when the lowest-cost feasible set can be found in polynomial time, a polynomial time separation oracle exists).

This auction appears to be very natural, and it is an obvious question to ask about what range of values it may give. It may appear that this restricted version could possibly give results that are within some restricted range between NTUmin and NTUmax. However, we will see examples that show it can attain NTUmax and may also be arbitrarily close to NTUmin (and even equal to NTUmin, when $\frac{\text{NTUmax}}{\text{NTUmin}} \leq 2$). Note that it can attain NTUmax exactly even in cases when this ratio is much larger than 2 (and proportional to n).

In Example 5.4 we can see that NTUmax is attainable by this process, and note that NTUmin<NTUmax (this result would be trivial if NTUmin and NTUmax were equal). This example uses the single-commodity auction, as described in Chapter 3.

Example 5.4.

This is a commodity auction with $\ell + 1$ agents in which we wish to purchase ℓ items. The quantity held by each agent A_e is given by q_e in Table 5.4. In this example, $\mathbf{b}^{\min}$ *denotes a* NTUmin *bid vector,* $\mathbf{b}^{\max}$ *is a* NTUmax *bid vector, and* $\mathbf{b}^{\uparrow}$ *is a uniformly ascending bid vector.*

	Agent	q_e	c_e	$b_e^{\min}$	$b_e^{\max}$	$b_e^{\uparrow}$
	A_1	1	0	2	1	1
	A_2	1	0	0	1	1
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	A_ℓ	1	0	0	1	1
	$A_{\ell+1}$	2	2			
Total				2	ℓ	ℓ

TABLE 5.4: Uniformly Ascending Bid $b^{\uparrow}$ Equals NTUmax

Observe that this example shows a Uniformly Ascending bid vector $\mathbf{b}^{\uparrow}$ *is reached when all of the bids have the same value. Recall that a Uniformly Ascending Bid vector satisfies conditions (2) and (3) in Definition 1.2. It is possible to verify that this bid vector does not violate condition (2); if it did then there would be some subset with quantity at most* 2 *that has a bid of greater than the cost of $A_{\ell+1} = 2$. It is equally possibly to verify that condition (3) is satisfied, as for every* 2 *agents in S, the sum of bids is exactly* 2.

As all the bids have increased by the same value, they have stopped ascending at the same time. Hence there can be no smaller value (i.e., $t' < 1$) that they could have stopped at, as if this smaller value would have satisfied condition (3) in Definition 1.2 then condition (2) in Definition 1.2 must have been violated by reaching $t = 1$, as all bids will be strictly greater than if they had stopped at t'.

This principle can be applied to some of the other examples of Uniformly Ascending bids that will be shown. Given a bid vector **b** *satisfying conditions (2) and (3), when all bids have risen from their cost by an equal value, then* **b** *is the vector that would be returned by the Uniformly Ascending auction.*

Example 5.5 shows that this process may obtain the value NTUmin, when NTUmin < NTUmax. Note, however, that this example does not easily generalize to instances when NTUmax > 2NTUmin. When NTUmax> 2NTUmin, for a single-commodity auction, then we do not know of any examples where the uniformly ascending process can give a value exactly as low as NTUmin.

Example 5.5. *A commodity auction with* 5 *agents where we wish to purchase* 5 *identical items. Each agent* $A_e \in \{A_1, A_2, A_3, A_4, A_5\}$ *has the quantity* q_e *and cost* c_e *given in Table 5.5. In this example,* $\mathbf{b}^{\min}$ *denotes a* NTUmin($\mathbf{c}$) *bid vector,* $\mathbf{b}^{\max}$ *is a* NTUmax($\mathbf{c}$) *bid vector, and* $\mathbf{b}^{\uparrow}$ *is a uniformly ascending bid vector.*

	Agent	q_e	c_e	$b_e^{\min}$	$b_e^{\max}$	$b_e^{\uparrow}$
	A_1	1	0	1	0	1
	A_2	2	0	1	2	1
	A_3	2	0	1	2	1
	A_4	1	1			
	A_5	3	2			
Total				3	4	3

TABLE 5.5: Uniformly Ascending Bid $b^{\uparrow}$ equals NTUmin

Observe that every $b_e^{\uparrow}$ *bid has the same increase in value from the cost, and can be verified to satisfy condition (2) and (3) in Definition 1.2. Hence* $\mathbf{b}^{\uparrow}$ *is a valid Uniformly Ascending Bid.*

It is also the case that a value close to NTUmin (again, when NTUmin < NTUmax) can be obtained when there is no restriction on the ratio between NTUmax and NTUmin. As a brief description of how this may be achieved, if we consider some example when NTUmin and NTUmax may differ by a large factor, there will be some agent(s) that receive a relatively large bid in an NTUmin vector. By using many agents to 'simulate' the effect of these particular agent(s) (that would command a large bid) then as the bidding rises uniformly, the sum of the bids for these many agents will rise quickly. As other agents, which would receive a lower NTUmin bid, will rise much more slowly in comparison (not being simulated by multiple agents), then we will end the uniformly ascending process with a bid vector with a value close to NTUmin (introducing the extra agents will not affect the value of NTUmin).

In a commodity auction this can be done by having many agents with a small quantity replace a single agent with a large quantity, and we will see examples of how this may occur.

Recall that Example 5.4, shows a single-commodity auction showing a large difference (a factor of $\ell/2$) between NTUmin and NTUmax, when the ascending price is equal to NTUmax. As an intermediate step, we will scale up all of the quantities, by some constant factor k, to give Example 5.6.

Example 5.6.

This is a commodity auction with $\ell + 1$ agents to purchase ℓk identical items. Each agent $A_e \in \{A_1, \ldots, A_{\ell+1}\}$ has the quantity q_e and cost c_e given in Table 5.6. In this example, $\mathbf{b}^{\min}$ *denotes a* NTUmin *bid vector,* $\mathbf{b}^{\max}$ *is a* NTUmax *bid vector, and* $\mathbf{b}^{\uparrow}$ *is a uniformly ascending bid vector.*

	Agent	q_e	c_e	$b_e^{\min}$	$b_e^{\max}$	$b_e^{\uparrow}$
	A_1	k	0	2	1	1
	A_2	k	0	0	1	1
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	A_ℓ	k	0	0	1	1
	$A_{\ell+1}$	$2k$	2			
Total				2	ℓ	ℓ

TABLE 5.6: Example 5.4 Multiplied by constant k

Observe that every $b_e^{\uparrow}$ bid has the same increase in value from the cost, and can be verified to satisfy conditions (2) and (3). Hence $\mathbf{b}^{\uparrow}$ *is a valid Uniformly Ascending bid vector.*

In Example 5.7 we then remove the single agent A_1 (when $q_1 = k$) and replace it with k other agents, $A'_1, \ldots, A'_k$ (when $q_{\{A'_1,\ldots,A'_k\}} = k$). NTUmin and NTUmax are unchanged, but the ascending price auction now gives a result much closer to NTUmin (specifically, it is $2 + 2(\ell - 1)/k$ which approaches 2 for large k).

Example 5.7. *This is a commodity auction with $k + \ell$ agents to purchase ℓk identical items. Each agent A'_e has the quantity q_e and cost c_e given in Table 5.7.*

In this example, $\mathbf{b}^{\min}$ *denotes a* NTUmin *bid vector,* $\mathbf{b}^{\max}$ *is a* NTUmax *bid vector, and* $\mathbf{b}^{\uparrow}$ *is a uniformly ascending bid vector.*

	Agent	q_e	c_e	$b_e^{\min}$	$b_e^{\max}$	$b_e^{\uparrow}$
	A'_1	1	0	2	$1/k$	$2/k$
	A'_2	1	0	0	$1/k$	$2/k$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	A'_k	1	0	0	$1/k$	$2/k$
	A_2	k	0	0	1	$2/k$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	A_ℓ	k	0	0	1	$2/k$
	$A_{\ell+1}$	$2k$	2			
Total				2	ℓ	$2+2(\ell-1)/k$

TABLE 5.7: Example 5.6 substituting A_1 with $A'_1, \ldots, A'_k$

Observe that the Uniformly Ascending bids are all equal, and can be verified to satisfy conditions (2) and (3). Hence $\mathbf{b}^{\uparrow}$ *is a valid Uniformly Ascending bid vector.*

5.8 Path Auctions

We will also examine the range of values that are possible in the well-known path auction (described in Chapter 4). In Figure 5.2 we see an example for a shortest path auction that shows the ascending price auction gives a value that is close to NTUmin. Note that the ratio shown of $\frac{\text{NTUmax}}{\text{NTUmin}} = 2$ is equal to the upper bound of $\frac{\text{NTUmax}}{\text{NTUmin}} \leq 2$ proven in [11]. This construction is a variation of the 'diamond graph' that was shown by Karlin et al. [24] and used by Elkind et al. [11] to demonstrate a lower bound of 2 on the ratio of NTUmax/NTUmin . In this example, the direct path shown between s and t (via u and v) has cost 0, and hence is chosen as the winning set. In order to satisfy condition (3) in Definition 1.2, the winning edges between s and v must have a sum of bids equal to 1. The winning edges between u and t must also have a sum of bids equal to 1. As the edges between u and v occur in both constraints, minimizing their sum of bids has the effect of maximizing the bid vector, and vice-versa.

As previously, in order to maximize the sum of bids between u and v, in a uniformly rising process we 'simulate' an edge between u and v with some large number, k, of separate edges. Hence there are $k+1$ edges between s and v and also between u and t. In a uniformly rising bid process, when each of these winning edges has a bid of $1/(k+1)$, condition (3) in Definition 1.2 is satisfied. (Figure 5.3 behaves similarly and Figure 5.4, instead, uses k edges to simulate the two outer edges of the winning path, resulting in higher bids.)

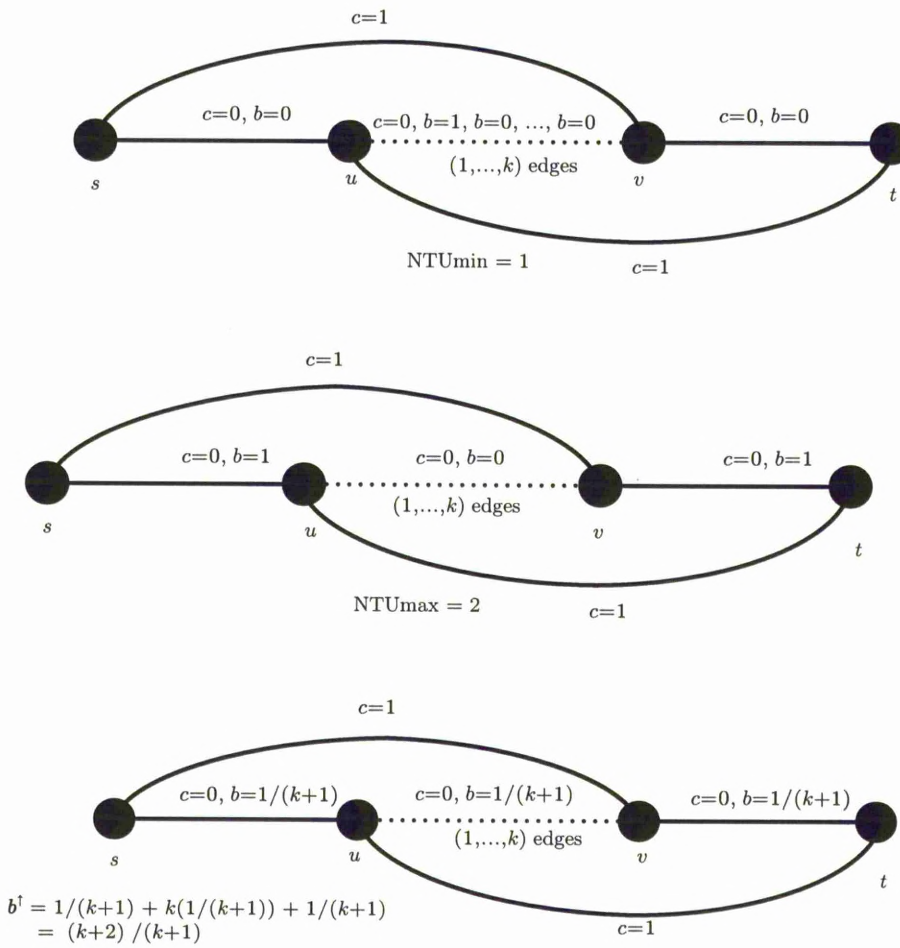


FIGURE 5.2: Uniformly Ascending Price Auction for Shortest Path Auction

Observe that the Uniformly Ascending bids are all equal, and can be verified to satisfy condition (2) and (3) of Definition 1.2.

Perhaps unsurprisingly, should the uniformly ascending-price auction start from bid values of zero, rather than the cost, a larger range of values is possible, as any bid vector obtained would have to satisfy condition (1*) in Definition 1.3 rather than condition (1) in Definition 1.2.

In Figure 5.3 we see an example when beginning from zero would produce a bid vector, $\mathbf{b}^{\uparrow}$ whose total is below NTUmin (but above TUmin). When we have NTUmin= 3 and TUmin = 2, we can have $b_S^{\uparrow} = 2 + \frac{2}{k+1}$ for any constant $k \geq 1$. In Figure 5.4 see that beginning from zero could produce a bid vector, $\mathbf{b}^{\uparrow}$ which is above NTUmax; we have NTUmax= 3, TUmax = 4, and we can have $b_S^{\uparrow} = 3 + \frac{k-1}{k+1}$ for any constant $k \geq 1$.

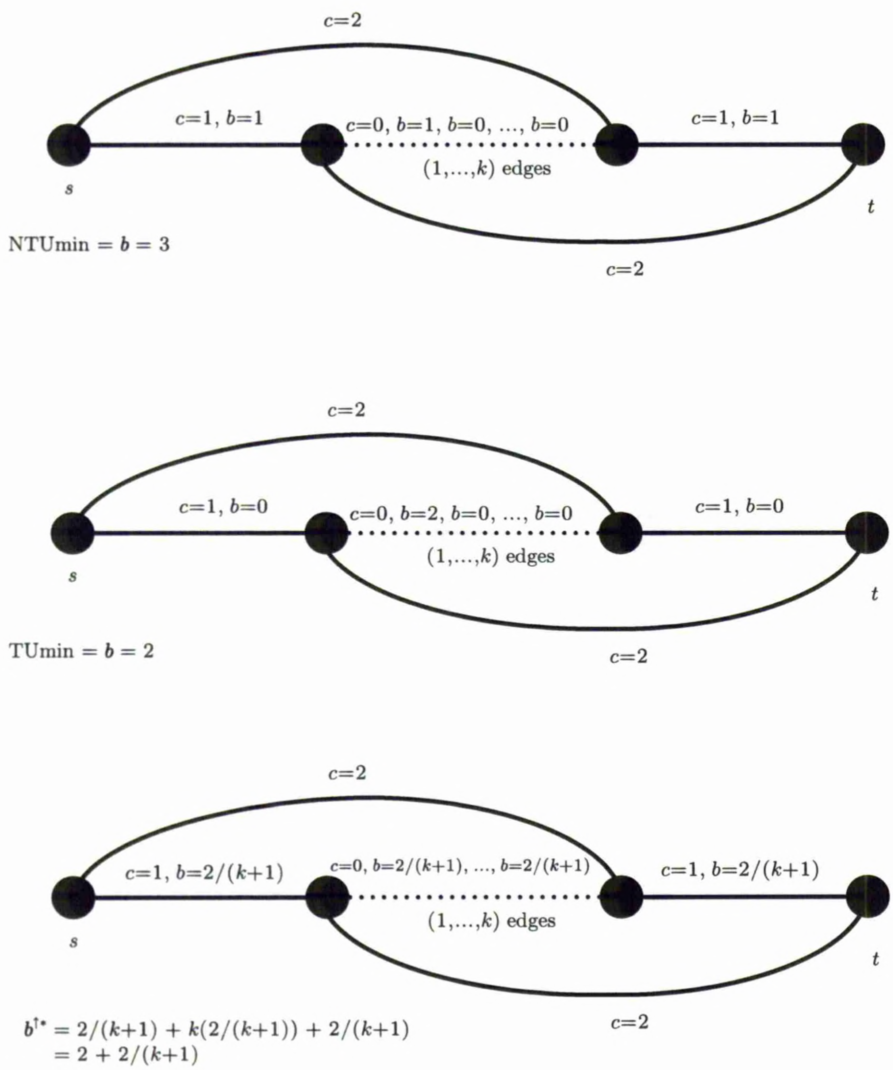


FIGURE 5.3: A Shortest Path Auction Where the Uniformly Ascending from Zero Price is Below NTUmin

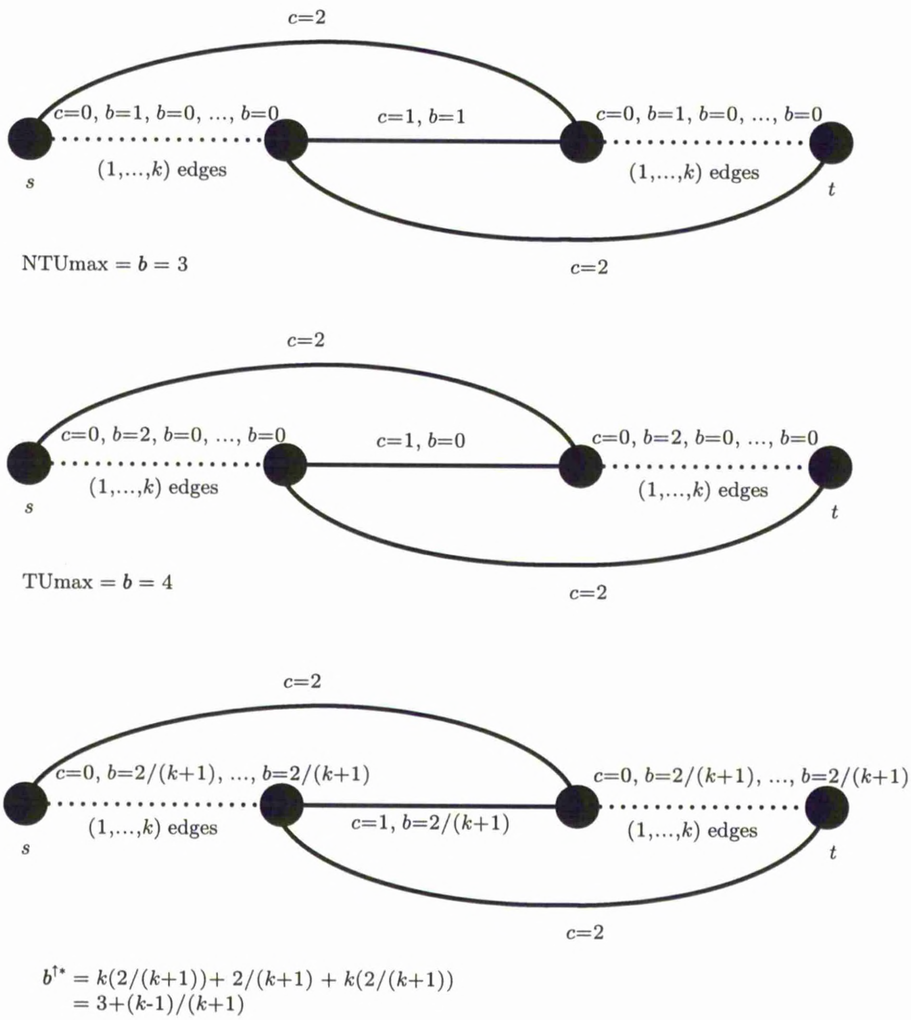


FIGURE 5.4: A Shortest Path Auction where the Uniformly Ascending from Zero Price is Above NTUmax

5.9 Ordered Maximal Bidding

In Sections 5.3 and 5.5 we saw the idea that agents may be required to make their bids in some order, with later bids depending on the value of the earlier bids. In those auctions, we required that the bid increased by some small increment. If we remove this restriction, and allow the agents to each choose their own bid, then it is obvious that any rational agent will simply make their bid as large as possible, such that they are still in the winning set. Therefore each agent bids exactly once. As no agent could gain greater utility by bidding again, later on, there is no loss of generality in restricting the ordering function to a single permutation of the winning set.

We will now examine this process for a first-price auction, which we will call 'Ordered Maximal Bidding'. In this process, given a specific ordering of the winning agents, each will, in turn, choose a bid to be as high as possible (until we have some T_e satisfying (3), giving a 'tight' constraint). Bid totals are always in the range between NTUmin and NTUmax, as any bid vector produced is feasible — all bids start from their cost, satisfying condition (1) in Definition 1.2, no agent can raise such that there would be a cheaper feasible set, satisfying condition (2) in Definition 1.2, and every agent will raise until it can bid no higher due to the presence of an alternative feasible set, thus satisfying condition (3) in Definition 1.2.

As in the previous first-price auctions, we would like to understand what range of values are obtainable from this process. For example, can we always choose an ordering that may result in NTUmin? If not, can we achieve a bid total that is within some constant factor of NTUmin? Would this hold even for restricted settings?

5.9.1 Definitions

We will fix a winning set S (let $m = |S|$ and $S = \{e_1, \ldots, e_m\}$), and a cost vector $\mathbf{c}$. Let σ_S denote a permutation, or sequence, of the elements in S. During this section, we will fix S as the lexicographically first winning set (amongst those with lowest cost), and to simplify the notation, let σ_S be denoted by σ. For brevity, when σ^α denotes a specific ordering on S, let $\mathbf{b}^\alpha$ be the ordered maximal bid vector for that ordering, rather than the more clumsy notation $\mathbf{b}^{\sigma^\alpha}$. Likewise, let $\mathbf{c}^\alpha$ be the vector of costs, given the ordering σ^α.

If $\sigma = (e'_1, \ldots, e'_m)$ is some ordering of S then define the resulting bid vector $\mathbf{b}^\sigma = (b^\sigma_1, \ldots, b^\sigma_m)$ as follows:

$$b^\sigma_1 = \max\{b_1 : (b_1, c_{e'_2}, \ldots, c_{e'_m}) \text{ satisfies (2)}\}$$
$$b^\sigma_2 = \max\{b_2 : (b^\sigma_1, b_2, c_{e'_3}, \ldots, c_{e'_m}) \text{ satisfies (2)}\}$$
$$b^\sigma_3 = \max\{b_3 : (b^\sigma_1, b^\sigma_2, b_3, c_{e'_4}, \ldots, c_{e'_m}) \text{ satisfies (2)}\}$$
$$\vdots$$
$$b^\sigma_m = \max\{b_m : (b^\sigma_1, \ldots, b^\sigma_{m-1}, b_m) \text{ satisfies (2)}\}$$

in general, for $i = (1, \ldots, m)$ (in order)

$$b^\sigma_i = \max\{b_i : (b^\sigma_1, \ldots, b^\sigma_{i-1}, b_i, c_{e'_{i+1}}, \ldots, c_{e'_m}) \text{ satisfies (2)}\}$$

Let $b^\sigma_S = \sum_{i \in S} b^\sigma_i$ be the sum of bids for the winning set S, and now define the minimum and maximal bids that are obtainable with a maximal ordered bid. Let

$$\text{OMBmin}(\mathbf{c}) = \min_\sigma b^\sigma_S$$

and let

$$\text{OMBmax}(\mathbf{c}) = \max_\sigma b^\sigma_S.$$

We will now see some examples of the ordered maximal bid process, and the values that result.

5.9.2 Examples

In Example 5.8 we see that the choice of permutation can give either NTUmin, NTUmax, or some other value which is between the two. This example is presented as a single-commodity auction where 9 items are desired.

Example 5.8. *Suppose there are* 8 *agents, each* $A_e \in \{A_1, \ldots, A_8\}$ *with a quantity* q_e *and cost* c_e *given in Table 5.8. Assume the buyer desires* 9 *units.*

	Agent	q_e	c_e	b_e^α	b_e^β	b_e^γ
	A_1	1	0	3	2	2
	A_2	1	0	0	1	1
	A_3	2	0	2	3	2
	A_4	2	0	2	3	2
	A_5	3	0	4	4	5
	A_6	2	3			
	A_7	3	5			
	A_8	5	7			
Total				11	13	12

TABLE 5.8: Ordered Maximal Bid Auction for 9 items giving differing results

The winning set is $S = \{A_1, A_2, A_3, A_4, A_5\}$ *and the permutations used are* $\sigma^\alpha = (A_1, A_2, A_3, A_4, A_5)$, $\sigma^\beta = (A_3, A_4, A_1, A_2, A_5)$, *and* $\sigma^\gamma = (A_5, A_1, A_2, A_3, A_4)$ *giving the ordered maximal bid vectors* $b_e^\alpha, b_e^\beta, b_e^\gamma$ *respectively.*

Let $\mathbf{b}^\alpha$ *be the bid vector calculated from the ordering* σ^α, *and this has bids increased in order depending on the quantity of each agent* $(q_e = 1, q_e = 2, q_e = 3)$.

Let $\mathbf{b}^\beta$ *be the bid that uses* σ^β, *and is in order* $(q_e = 2, q_e = 1, q_e = 3)$ *and* $\mathbf{b}^\gamma$ *uses* σ^γ *in order* $(q_e = 3, q_e = 1, q_e = 2)$. *We will now see exactly how the bid vectors were calculated for each of these orderings; Procedure 1 describes the process for ordering* σ^α, *giving one of the possible tight constraints at each step, and showing how the bid value is determined from that tight constraint as well as the previously determined bids. Likewise, Procedure 2 describes the ordering* σ^β *and Procedure 3 the ordering* σ^γ. *Order* σ^α *gives total* $b_S^\alpha = 11$, *order* σ^β *gives* $b_S^\beta = 13$, *and order* σ^γ *gives* $b_S^\gamma = 12$.

Procedure 1 Calculating Bids for $\mathbf{b}^\alpha$

b_1^α is restricted by $b_1^\alpha <= b_6^\alpha$	$b_1^\alpha = b_6^\alpha = 3$
b_2^α is restricted by $b_1^\alpha + b_2^\alpha <= b_6^\alpha$	$b_2^\alpha = (b_6^\alpha - b_1^\alpha) = (3 - 3) = 0$
b_3^α is restricted by $b_1^\alpha + b_3^\alpha <= b_7^\alpha$	$b_3^\alpha = b_7^\alpha - b_1^\alpha = 5 - 3 = 2$
b_4^α is restricted by $b_1^\alpha + b_4^\alpha <= b_7^\alpha$	$b_4^\alpha = b_7^\alpha - b_1^\alpha = 5 - 3 = 2$
b_5^α is restricted by $b_1^\alpha + b_5^\alpha <= b_8^\alpha$	$b_5^\alpha = b_8^\alpha - b_1^\alpha = 7 - 3 = 4$

Procedure 2 Calculating Bids for $\mathbf{b}^\beta$

b_3^β is restricted by $b_3^\beta <= b_6^\beta$	$b_3^\beta = b_6^\beta = 3$
b_4^β is restricted by $b_4^\beta <= b_6^\beta$	$b_4^\beta = (b_6^\beta) = 3$
b_1^β is restricted by $b_3^\beta + b_1^\beta <= b_7^\beta$	$b_1^\beta = (b_7^\beta - b_3^\beta) = (5 - 3) = 2$
b_2^β is restricted by $b_1^\beta + b_2^\beta <= b_6^\beta$	$b_2^\beta = (b_6^\beta - b_1^\beta) = (3 - 2) = 1$
b_5^β is restricted by $b_3^\beta + b_5^\beta <= b_8^\beta$	$b_5^\beta = (b_8^\beta - b_3^\beta) = (7 - 3) = 4$

Procedure 3 Calculating Bids for $\mathbf{b}^\gamma$

b_5^γ is restricted by $b_5^\gamma <= b_7^\gamma$	$b_5^\gamma = b_7^\gamma = 5$
b_1^γ is restricted by $b_5^\gamma + b_1^\gamma <= b_8^\gamma$	$b_1^\gamma = (b_8^\gamma - b_5^\gamma) = (7 - 5) = 2$
b_2^γ is restricted by $b_1^\gamma + b_2^\gamma <= b_8^\gamma$	$b_2^\gamma = (b_6^\gamma - b_1^\gamma) = (3 - 2) = 1$
b_3^γ is restricted by $b_5^\gamma + b_3^\gamma <= b_8^\gamma$	$b_3^\gamma = (b_8^\gamma - b_5^\gamma) = (7 - 5) = 2$
b_4^γ is restricted by $b_5^\gamma + b_4^\gamma <= b_8^\gamma$	$b_4^\gamma = (b_8^\gamma - b_5^\gamma) = (7 - 5) = 2$

Random Orderings

Theorem 5.4. *There exists a sequence of set-system auctions indexed by n, where* NTUmax/ NTUmin $\geq n - 2$, *such that, for any ordering σ that is chosen uniformly at random, the bid b_S^σ approaches* NTUmax *in expectation.*

Proof. In Example 5.9 we see that any ordered maximal auction, choosing a sequence uniformly at random, gives an expected value of $\frac{\text{NTUmin}+(\ell-1)\text{NTUmax}}{\ell}$ which is close to NTUmax for large ℓ (and hence large n). □

Example 5.9.

This is a commodity auction with $\ell + 1$ agents to purchase $2\ell + 1$ identical items. Each agent $A_e \in \{A_1, \ldots, A_{\ell+1}\}$ has the quantity q_e and cost c_e given in Table 5.9.

	Agent	q_e	c_e	$b_e^{\min}$	$b_e^{\max}$
	A_1	1	0	1	0
	A_2	2	0	0	1
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
	A_ℓ	2	0	0	1
	$A_{\ell+1}$	3	1		
Total				1	$\ell-1$

TABLE 5.9: Ordered Maximal Bid Auction, choosing randomly

We see that any ordered maximal auction, choosing a sequence uniformly at random, will select A_1 first on $1/\ell$ occasions and any other agent on $(\ell-1)/\ell$ occasions. It is only by choosing A_1 first that NTUmin *will be obtained, otherwise* NTUmax *will be obtained.*

Bounds for Values Achievable with Ordered Bidding

In Example 5.10 we see that **any** ordered auction will result in NTUmin being selected, whereas NTUmax is not achievable, regardless of the order (i.e., OMBmax < NTUmax).

Example 5.10. *This is a commodity auction with $\ell+1$ agents to purchase ℓ identical items. Each agent $A_e \in \{A_1, \ldots, A_{\ell+1}\}$ has the quantity q_e and cost c_e given in Table 5.10.*

	Agent	q_e	c_e	$b_e^{\min}$	$b_e^{\max}$	b_e^σ
	A_1	1	0	1	1/2	1
	A_2	1	0	0	1/2	0
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
	A_ℓ	1	0	0	1/2	0
	$A_{\ell+1}$	2	1			
Total				1	$\ell/2$	1

TABLE 5.10: Ordered Maximal Bid Auction may not find NTUmax

Any agent e that appears first in the ordering will have $b_e^\sigma = 1$, and so every other agent $j \neq e$ must bid $b_j^\sigma = 0$. Hence, by this symmetry, OMBmax $= 1$.

However, Example 5.11 shows that NTUmin may not be achieved by any ordering (OMBmin > NTUmin), even with as few as 5 agents in S.

Example 5.11. *In this auction there are 7 agents, $\{A_1, \ldots, A_7\}$ with feasible sets as follows.*

$\mathcal{F} = \{\{A_1, A_2, A_3, A_4, A_5\}, \{A_6, A_3, A_4, A_5\}, \{A_6, A_1, A_4, A_5\}, \{A_6, A_2, A_4, A_5\}, \{A_7, A_4\}, \{A_7, A_5\}\}$. *The cost c_e for each agent A_e is given in Table 5.11. Observe that the winning set* $S = \{A_1, A_2, A_3, A_4, A_5\}$.

	Agent	c_e	$b_e^{\min}$	b_e^{σ}
	A_1	0	1/2	1
	A_2	0	1/2	0
	A_3	0	1/2	0
	A_4	0	1/2	1
	A_5	0	1/2	1
	A_6	1		
	A_7	2		
Total			5/2	3

TABLE 5.11: Ordered Maximal Bid Auction may not find NTUmin when $|S| = 5$

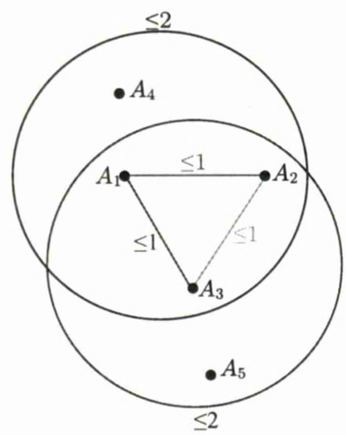


FIGURE 5.5: Hypergraph Representation of constraints for Table 5.11

To verify that this has OMBmin ≥ 3, *we can examine all possible orderings; when A_1 appears first in the ordering, then $b_1 = 1$ and the feasible set $\{A_6, A_3, A_4, A_5\}$ would mean that $b_2 \leq 0$, and the feasible set $\{A_6, A_2, A_4, A_5\}$ would imply $b_3 \leq 0$. Then we will have both $b_4 = 1$ and $b_5 = 1$ (as only the sets $\{A_7, A_5\}$ and $\{A_7, A_4\}$ respectively are alternative solutions for A_5 and A_4) giving* OMBmin ≥ 3.

Similarly, when A_2 appears first then $b_2 = 1$ and the feasible sets $\{A_6, A_3, A_4, A_5\}$ and $\{A_6, A_1, A_4, A_5\}$ would give $b_1 = b_3 = 0$ and hence $b_4 = b_5 = 1$ and OMBmin≥ 3. *When A_3 appears first, $b_3 = 1$ and the feasible sets $\{A_6, A_2, A_4, A_5\}$ and $\{A_6, A_1, A_4, A_5\}$ would give $b_1 = b_2 = 0$ and hence $b_4 = b_5 = 1$ and* OMBmin≥ 3. *Now, consider that when A_4 appears first, then $b_4 = 2$, and the set $\{A_7, A_5\}$ would give $b_1 = b_2 = b_3 = 0$,* *and hence* $b_5 = 2$ *and* OMBmin≥ 4. *Finally, when* A_5 *appears first, then* $b_5 = 2$, *and the set* $\{A_7, A_4\}$ *would give* $b_1 = b_2 = b_3 = 0$, *and hence* $b_4 = 2$ *and* OMBmin≥ 4.

Example 5.12 also shows an example where OMBmin > NTUmin, with $|S| = 6$. We present this example primarily as a precursor to a more general result. The structure of this example will be generalized, in Example 5.13,to give a lower bound of $\Omega(n)$ for the ratio of OMBmin/NTUmin.

Example 5.12. *In this auction there are 7 agents,* $\{A_1, \ldots, A_7\}$ *with feasible sets as follows.*

$\mathcal{F} = \{\{A_1, A_2, A_3, A_4, A_5, A_6\}, \{A_7, A_3, A_5, A_6\}, \{A_7, A_2, A_4, A_6\}, \{A_6, A_2, A_4, A_5\}, \{A_7, A_1, A_4, A_5\}\}$. *The cost* c_e *for each agent* A_e *is given in Table 5.12. Observe that the winning set* $S = \{A_1, A_2, A_3, A_4, A_5, A_6\}$.

	Agent	c_e	$b_e^{\min}$	b_e^{σ}
	A_1	0	1/2	1
	A_2	0	1/2	0
	A_3	0	1/2	0
	A_4	0	0	0
	A_5	0	0	0
	A_6	0	0	1
	A_7	1		
Total			3/2	2

TABLE 5.12: Ordered Maximal Bid Auction May Not Find NTUmin

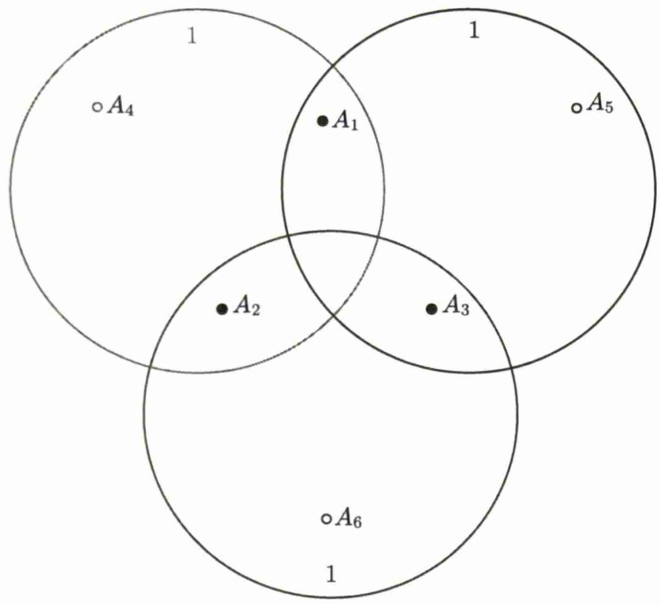


FIGURE 5.6: Hypergraph Representation of Constraints for Table 5.12

Theorem 5.5. *For any $n \geq 9$ there exists a set-system auction with cost vector* $\mathbf{c}$ *such that* $\text{OMBmin}(\mathbf{c})/\text{NTUmin}(\mathbf{c}) \geq 2n/9 - 2/3$.

Proof. In Example 5.13 we have some integer parameter $\ell > 0$. (While this may appear to be a complex example, it is simply a generalization of Example 5.12, which has $\ell = 1$). Here, we see that there exists a feasible bid vector such that $b_1 = b_2 = b_3 = 1/2$. As any two of these agents sum up to 1, $(b_1 + b_2 = b_2 + b_3 = b_1 + b_3 = 1)$ and as all constraints share at least two of these agents, no other agent may bid higher than zero, giving $\text{NTUmin}(\mathbf{c}) \leq 3/2$. Observe that any ordered bid vector must give some agent e a bid of $b_e = 1$ (i.e., agent e is first in the ordering). If $b_1 = 1$ then in order to satisfy the constraints on $b_1 + b_2$ and $b_1 + b_3$ we must have $b_2 = b_3 = 0$. There are ℓ agents that share a constraint only with A_2 and A_3 (namely agents $A_{4+\ell}, \ldots, A_{4+2\ell-1}$), and hence each of these agents may bid 1. Therefore that are at least $\ell + 1$ agents that will bid at least 1. By symmetry, observe that this would also be true if we had $b_2 = 1$ or $b_3 = 1$. If none of $\{A_1, A_2, A_3\}$ bid 1, then each agent has some neighbours that can bid 1, and so all other 3ℓ agents $(A_4, \ldots, A_{4+3\ell-1})$ will bid 1, hence we have $\text{OMBmin} \geq \ell + 1$. There are a total of $3\ell + 3$ agents, so we can give this lower bound in terms of n, the number of agents.

As ℓ must be an integer parameter, there is some suitable ℓ such that $n \leq 3\ell + 3 + 2$, and hence $\ell \geq (n-5)/3$. (If n is not divisible by 3, then we may add at most 2 agents to the example, that are in no feasible solutions, to give an instance of size n.) Starting with

$$\text{OMBmin}(\mathbf{c}) \geq \ell + 1$$

substitute the lower bound for ℓ

$$\text{OMBmin}(\mathbf{c}) \geq (n-5)/3 + 1$$

and rewrite to give

$$\text{OMBmin}(\mathbf{c}) \geq (n-2)/3$$

As we have $\text{NTUmin}(\mathbf{c}) \leq 3/2$ therefore we can rewrite to give

$$\text{OMBmin}(\mathbf{c})/\text{NTUmin}(\mathbf{c}) \geq 2n/9 - 2/3$$

□

Example 5.13. *In this example of a set-system auction we have a parameter* ℓ *and a total of* $3\ell+4$ *agents* $(A_0, \dots, A_{3\ell+3})$*. The feasible sets are given as follows.*

$$\begin{array}{llll} \mathcal{F} = \{\mathcal{E} \setminus \{A_0\}, & & & \\ \mathcal{E} \setminus \{A_1, A_2, A_4\}, & \mathcal{E} \setminus \{A_1, A_2, A_{4+1}\}, & \dots & \mathcal{E} \setminus \{A_1, A_2, A_{4+\ell-1}\}, \\ \mathcal{E} \setminus \{A_2, A_3, A_{4+\ell}\}, & \mathcal{E} \setminus \{A_2, A_3, A_{4+\ell+1}\}, & \dots & \mathcal{E} \setminus \{A_2, A_3, A_{4+2\ell-1}\}, \\ \mathcal{E} \setminus \{A_1, A_3, A_{4+2\ell}\}, & \mathcal{E} \setminus \{A_1, A_3, A_{4+2\ell+1}\}, & \dots & \mathcal{E} \setminus \{A_1, A_3, A_{4+3\ell-1}\}\} \end{array}$$

For each agent $e \in \mathcal{E}$ *the cost value* c_e *is given in Table 5.13, as is a bid value* $b_e^{\min}$ *for a* NTUmin *bid vector and a bid value,* b_e^{σ}*, for a minimum Ordered Maximal Bid vector.*

Agent	c_e	$b_e^{\min}$	b_e^{σ}
A_1	0	1/2	1
A_2	0	1/2	0
A_3	0	1/2	0
A_4	0	0	0
$\vdots$	0	0	0
$A_{4+\ell-1}$	0	0	0
$A_{4+\ell}$	0	0	1
$\vdots$	0	0	1
$A_{4+2\ell-1}$	0	0	1
$A_{4+2\ell}$	0	0	1
$\vdots$	0	0	1
$A_{4+3\ell-1}$	0	0	1
A_0	1		
Total		3/2	$\ell+1$

TABLE 5.13: Ordered Maximal Bid Auction May Be Much Higher than NTUmin

Observe that the winning set $S = \mathcal{E} \setminus \{A_0\}$.

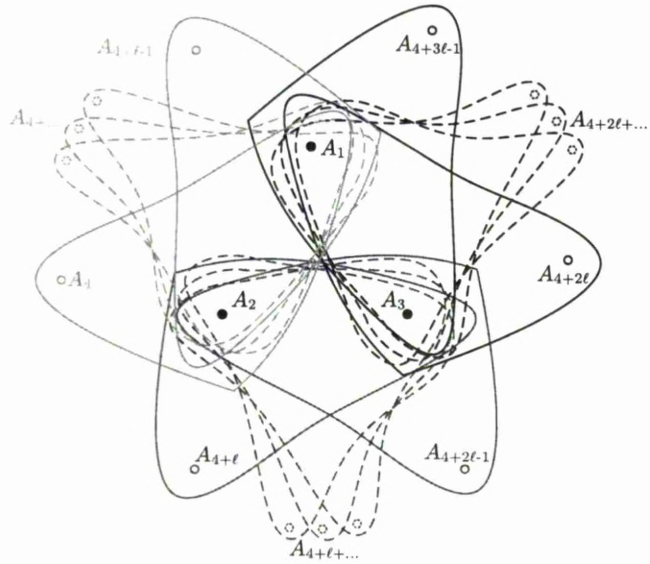


FIGURE 5.7: Hypergraph Representation of Constraints for Table 5.13

5.9.3 Restricted Setting

We will now consider the special case where every constraint is restricted to a single value (for simpler notation, we will assume this to be 1, but any other value would be equivalent by rescaling), and that the cost of the winning agents is $c_S = 0$. We will also impose an additional restriction that we will only have binary constraints — that is we will restrict the cardinality of the constraining sets to 2. More formally,

$$\forall T \in (\mathcal{F} \setminus \{S\}), |(S \setminus T)| = 2.$$

We will see, later on, that even in this restricted setting, calculating both NTUmin and OMBmin is NP-hard to calculate, and also NP-hard to approximate within a factor of $n^{1-\epsilon}$ for any constant $\epsilon > 0$.

One effect of the restriction to binary constraints is that the hypergraph representation of the constraints, that was described earlier, will now be a graph. Borrowing from this graph representation, we will call any two agents i and j 'neighbours' if they are share some common constraint . (i.e., $\exists T \in \mathcal{F}$ such that $S \setminus T = \{i, j\}$).

Let $N(i)$ be the set of neighbours of agent i. More formally, $j \in N(i) \Leftrightarrow \exists T \in \mathcal{F}$ such that $S \setminus T = \{i, j\}$.

In this setting, we will firstly see a lower bound on the ratio between OMBmin and NTUmin.

Example 5.14 shows that NTUmin < OMBmin even when the instance is restricted to binary constraints with a single value. (We will see that this setting is of importance later on).

Example 5.14. *In this auction there are* 10 *agents,* $\{A_1, \ldots, A_{10}\}$ *with feasible sets as follows.*

$\mathcal{F} = \{\{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9\}, \{A_{10}, A_3, A_4, A_5, A_6, A_7, A_8, A_9\},$
$\{A_{10}, A_2, A_4, A_5, A_6, A_7, A_8, A_9\}, \{A_{10}, A_1, A_4, A_5, A_6, A_7, A_8, A_9\},$
$\{A_{10}, A_2, A_3, A_5, A_6, A_7, A_8, A_9\}, \{A_{10}, A_2, A_3, A_4, A_6, A_7, A_8, A_9\},$
$\{A_{10}, A_1, A_3, A_4, A_5, A_7, A_8, A_9\}, \{A_{10}, A_1, A_3, A_4, A_5, A_6, A_8, A_9\},$
$\{A_{10}, A_1, A_2, A_4, A_5, A_6, A_7, A_9\}, \{A_{10}, A_1, A_2, A_4, A_5, A_6, A_7, A_8\}\}.$

The cost c_e *for each agent* A_e *is given in Table 5.14.*

	Agent	c_e	$b_e^{\min}$	b_e^{α}
	A_1	0	1/2	1
	A_2	0	1/2	0
	A_3	0	1/2	0
	A_4	0	1/2	0
	A_5	0	1/2	0
	A_6	0	1/2	1
	A_7	0	1/2	1
	A_8	0	1/2	1
	A_9	0	1/2	1
	A_{10}	1		
Total			9/2	5

TABLE 5.14: Ordered Maximal Bid Auction may not equal NTUmin in restricted setting

Observe that the winning set $S = \{A_1, \ldots, A_9\}$ *and assume an ordering* $\sigma^{\alpha} = (A_1, A_2, A_3, A_5, A_6, A_7, A_8, A_9)$ *resulting in bid vector* b_e^{α}.

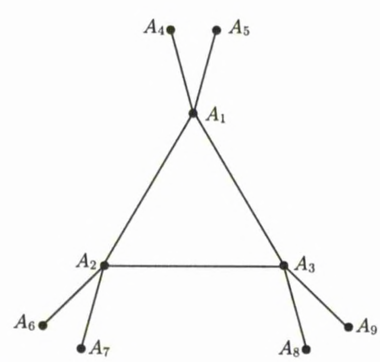


FIGURE 5.8: Graph Representation of Constraints in Table 5.14

The bid vector $\mathbf{b}^\sigma$ *shows the bids of one possible ordering. We will see that there is no other ordering possible that will give a lower total bid. Firstly, observe that every ordering* σ' *must give* $b_e^{\sigma'} \in \{0,1\}$ *for all* $e \in S$*. Secondly, any ordering that gives a* $b_1^{\sigma'} = 1$ *to agent* A_1 *must give* $b_2^{\sigma'} = b_3^{\sigma'} = 0$ *(as* $b_1^{\sigma'} + b_2^{\sigma'} \leq 1$ *and* $b_1^{\sigma'} + b_3^{\sigma'} \leq 1$*). In order to make the bids tight with regard to some constraint, then* $b_6^{\sigma'} = b_7^{\sigma'} = b_8^{\sigma'} = b_9^{\sigma'} = 1$ *giving* $b_S^{\sigma'} \geq 5$*. The constraint sets are symmetrical, so similar results for* $b_2^{\sigma'} = 1$ *and* $b_3^{\sigma'} = 1$ *are easily observed. Alternatively,* $b_1^{\sigma'} = b_2^{\sigma'} = b_3^{\sigma'} = 0$ *would clearly give* $b_S^{\sigma'} \geq 6$*, as all other agents would bid* 1 *to make some constraint tight.*

By generalizing Example 5.14 we are able to prove that there exists a class of auctions for which the ratio between OMBmin and NTUmin can approach 2.

Theorem 5.6. *For every small constant* $0 < \epsilon < 1$*, there exists a set-system auction, which is restricted to binary constraints with a single value, with cost vector* $\mathbf{c}$ *such that* $\mathrm{OMBmin}(\mathbf{c})/\mathrm{NTUmin}(\mathbf{c}) \geq 2 - \epsilon$*.*

Proof. Consider Example 5.15, with some integer parameter $\ell > 0$. This is, perhaps, more easily seen in Figure 5.9. This consists of a central clique, of size ℓ, (Agents $A_{0,1}, \ldots, A_{0,\ell}$) and each of these agent shares a constraint with an additional ℓ agents. There exists a bid vector, $\mathbf{b}$, when each agent may bid 1/2, giving a total of $\mathbf{b}^\sigma = (\ell^2 + \ell)/2$ and hence NTUmin $\leq (\ell^2 + \ell)/2$ — there are ℓ agents in the central clique, and ℓ agents attached to each of them, hence $\ell^2 + \ell$ total agents.

We will now examine possible ordered bid vectors, as two cases;

Case 1: There is some i such that $b_{0,i} = 1$.

Some agent $A_{0,i}$ in the central clique may bid $b_{0,i} = 1$, which results in the other $\ell - 1$ agents in the clique bidding 0 ($b_{0,j} = 0$ when $i \neq j$). As an agent $A_{0,j}$ in the clique is attached to ℓ other agents ($A_{j,1}, \ldots, A_{j,\ell}$), by sharing some constraint, there are $\ell(\ell-1)$ other agents that must bid ($b_{j,1} = 1, \ldots, b_{j,\ell} = 1$), and hence $\ell^2 - \ell + 1$ in total, including the single agent in the clique who bids 1.

Case 2: There is no i such that $b_{0,i} = 1$.

As no agent in the central clique bids 1, all ℓ^2 outside agents may bid 1, ($b_{x,y} = 1$ when $x \geq 1, y \geq 1$) hence a total bid of ℓ^2.

For a large value of ℓ, both ℓ^2 and $\ell^2 - \ell + 1$ will approach ℓ^2, and hence the ratio of OMBmin/NTUmin approaches $\ell^2/(\ell^2/2)$. Therefore, for the small constant ϵ, there is some large value of ℓ such that $\frac{2\ell^2 - \ell + 1}{\ell^2 + \ell} \geq 2 - \epsilon$. □

Example 5.15.

In this example of a set-system auction (with binary, single-valued constraints) we have a parameter ℓ *and a total of* $\ell^2 + \ell + 1$ *agents (*$A_{x,y} \in \mathcal{E}$ *for every* $0 \leq x \leq \ell$ *and* $1 \leq y \leq \ell$ *and also agent* $A_0 \in \mathcal{E}$*). The feasible sets are given as follows.*

$$
\begin{array}{lll}
\mathcal{F} = \{\mathcal{E} \setminus \{A_0\}, & & \\
\mathcal{E} \setminus \{A_{0,1}, A_{0,1}\}, & \ldots, & \mathcal{E} \setminus \{A_{0,1}, A_{0,\ell}\} \\
\vdots & \ddots & \vdots \\
\mathcal{E} \setminus \{A_{0,\ell}, A_{0,1}\}, & \ldots, & \mathcal{E} \setminus \{A_{0,\ell}, A_{0,\ell}\} \\
\mathcal{E} \setminus \{A_{0,1}, A_{1,1}\}, & \ldots, & \mathcal{E} \setminus \{A_{0,1}, A_{1,\ell}\} \\
\vdots & \ddots & \vdots \\
\mathcal{E} \setminus \{A_{0,\ell}, A_{\ell,1}\}, & \ldots, & \mathcal{E} \setminus \{A_{0,\ell}, A_{\ell,\ell}\}\}
\end{array}
$$

The cost c_e for each agent $e \in \mathcal{E}$ is given in Table 5.15, along with a bid value $b_e^{\min}$ for an NTUmin *bid vector and a minimum Ordered Maximal bid value b_e^{σ}.*

	Agent	c_e	$b_e^{\min}$	b_e^{σ}
	$A_{0,1}$	0	1/2	1
	$\vdots$	0	1/2	0
	$A_{0,\ell}$	0	1/2	0
	$A_{1,1}$	0	1/2	0
	$\vdots$	0	1/2	0
	$A_{1,\ell}$	0	1/2	0
	$\vdots$	0	1/2	1
	$\vdots$	0	1/2	1
	$\vdots$	0	1/2	1
	$A_{\ell,1}$	0	1/2	1
	$\vdots$	0	1/2	1
	$A_{\ell,\ell}$	0	1/2	1
	A_0	1		
Total			$(\ell^2+\ell)/2$	$\ell^2-\ell+1$

TABLE 5.15: OMBmin approaches 2 NTUmin

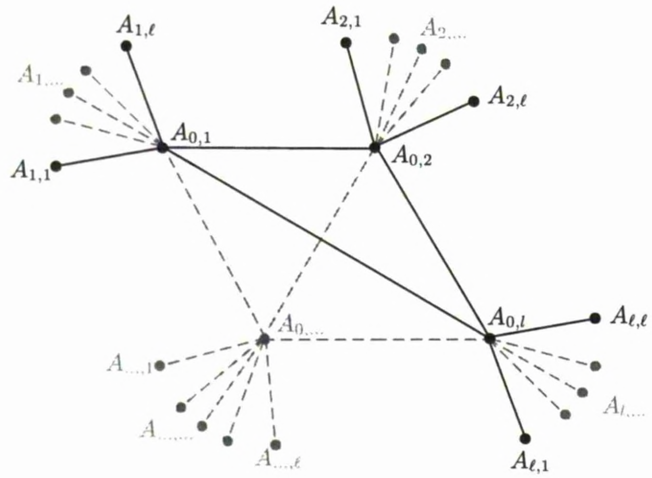


FIGURE 5.9: Graph Representation of Constraints in Table 5.15

Single-Commodity Auctions

As we have spent some time studying single-commodity auctions, it might also be interesting to study the ratio between OMBmin and NTUmin in that simple setting. We show here a lower bound for this ratio, again approaching 2.

Theorem 5.7. *For every small constant* $0 < \epsilon < 1$*, there exists a single-commodity auction with cost vector* $\mathbf{c}$ *such that* $\text{OMBmin}(\mathbf{c})/\text{NTUmin}(\mathbf{c}) \geq 2 - \epsilon$.

Proof. We will consider Example 5.16, with ℓ as some integer parameter.

Example 5.16. *This is a commodity auction with* $\ell + 7$ *agents to purchase* $2\ell + 7$ *identical items. Each agent* $A_e \in \{A_1, \dots, A_{\ell+7}\}$ *has the quantity* q_e *and cost* c_e *given in Table 5.16.*

Agent	q_e	c_e	$b_e^{\min}$	b_e^{α}	b_e^{β}
A_1	1	0	1/2	0	1
A_2	2	1	3/2	2	1
A_3	2	1	3/2	1	1
A_4	2	0	1/2	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$		
$A_{4+\ell}$	2	0	1/2	1	1
$A_{5+\ell}$	3	2			
$A_{6+\ell}$	4	3			
$A_{7+\ell}$	7	4			
Total			$(\ell+8)/2$	$\ell+4$	$\ell+4$

TABLE 5.16: OMBmin approaches 2 NTUmin for Single-Commodity Auctions

Observe that the bid vector $\mathbf{b}^{\min}$ *given is a feasible bid vector, and hence* NTUmin$(\mathbf{c})$ $\leq (\ell+8)/2$. *To verify this satisfies condition (3) in Definition 1.2, let* $e \in \{4, \dots, 4+\ell\}$ *and let* $T_e = S \setminus \{A_1, A_2, A_3, A_e\} \cup \{A_{7+\ell}\}$, *which gives* $b_{\{A_1,A_2,A_3,A_e\}} = 4$, *so* T_e *satisfies condition (3) for all* $A_1, \dots, A_{4+\ell}$. *Less formally, no agent can raise its bid due to the constraint implied by a replacement subset of quantity 7, with cost 4.*

We consider the ordered maximal bid process on Example 5.16. Let e be the first agent in the ordering σ, *and we will examine this as two cases.*

Case 1: $q_e = 1$.

Agent A_1 *may raise first and will be constrained by the bid of* $A_{5+\ell}$ *— this can be written as* $b_1^\sigma + b_2^\sigma \leq 2$. *This gives* $b_1^\sigma \leq 2 - c_2$, *and as* $c_2 = 1$, *therefore* $b_1^\sigma \leq 1$. *Once we have* $b_1^\sigma = 1$, *then the same constraint gives* $b_1^\sigma + b_j^\sigma \leq 2$ *for all* $j \in \{2, \dots, 4+\ell-1\}$, *and all other agents will bid* 1. *This gives each agent* $e \in S$ *a bid* $b_e^\sigma = 1$, *and therefore* $b_S^\sigma = |S| = \ell + 4$.

Case 2: $q_e = 2$.

If the first agent in the ordering e has $q_e = 2$, *then this will be constrained by both* $A_{5+\ell}$ *and* $A_{6+\ell}$. *As* $A_{5+\ell}$ *gives* $b_e^\sigma + b_1^\sigma \leq 2$, *and* $b_1^\sigma = 0$ *(as we have* $b_1^\sigma = c_1$ *until we reach agent* A_1 *in the ordering) then we have* $b_e^\sigma = 2$. *By the same constraint (from* $A_{5+\ell}$*), this will give* $b_1^\sigma = 0$. *As* $A_{6+\ell}$ *will constrain* $b_e^\sigma + b_j^\sigma$ *for all* $j \neq e, j > 1$ *we will have* $b_j^\sigma = 1$ *for all such j. This gives one agent e, a bid of* $b_e^\sigma = 2$, *agent* 1 *a bid of* $b_1^\sigma = 0$ *and every other agent j has* $b_j^\sigma = 1$. *Therefore we have* $b_S^\sigma = |S| = \ell + 4$.

In Example 5.16, we have seen that every ordered bid vector $\mathbf{b}^\sigma$ has $b_S^\sigma = \ell + 4$, and we have seen NTUmin$(\mathbf{c}) \leq (\ell + 8)/2$. Hence, for any small ϵ then there is some large value of ℓ such that $\frac{2(\ell+4)}{\ell+8} \geq 2 - \epsilon$.

□

This has shown that single-commodity auctions may give $\frac{\text{OMBmin(c)}}{\text{NTUmin(c)}} \geq 2 - \epsilon$ for some small value of $\epsilon < 1$. In the general case, we have seen that this ratio is $\Omega(n)$, so it is an obvious open question whether ratios of larger than 2 are possible for single-commodity auctions.

In Example 5.16, we saw that OMBmin approaches 2×NTUmin for single-commodity auctions, but we have not seen any instances that must give ratios of larger than 2.

We are, therefore, going to look at some seemingly reasonable approaches to choosing an ordering in order to determine what ratios they may give with respect to NTUmin. Example 5.9 shows that choosing a random ordering may give an expected value that is close to NTUmax, even when this is close to a factor of n larger than NTUmin.

In looking for an ordering that gives OMBmin close to NTUmin, one approach that may seem reasonable is to partition S into subsets of a given quantity, then choose the ordering simply based on the cardinality of these partitions. Example 5.17 and Example 5.18 show that this would be unsuccessful in always finding a bid vector close to NTUmin. In Example 5.17, a minimum value would be obtained by soliciting bids from the agents with the lower cardinality first (i.e., there are fewer agents with quantity ℓ, so they are ordered first). Example 5.18 shows that a minimum value would only be obtained by soliciting bids from the agents with the higher cardinality first. Both examples show that choosing the order incorrectly gives a bid vector **b** that is much higher than optimal. We have $b^\sigma_S \geq \ell - 1$ and NTUmin $= 1$, and as $\ell = (n/2) - 1$ then $\frac{b^\sigma_S}{\text{NTUmin}} \geq \Omega(n)$.

Example 5.17. *This is a single-commodity auction with $2\ell + 2$ agents to purchase $\ell^2 + \ell + 1$ identical items. Each agent $A_e \in \{A_1, \ldots, A_{2\ell+2}\}$ has the quantity q_e and cost c_e given in Table 5.17*

	Agent	q_e	c_e	$b_e^{\min}$	b_e^σ
	A_1	ℓ	0	0	1
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	A_ℓ	ℓ	0	0	1
	$A_{\ell+1}$	1	0	1	0
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$A_{2\ell+1}$	1	0	0	0
	$A_{2\ell+2}$	$\ell+1$	1		
Total				1	$\ell-1$

TABLE 5.17: Ordered Maximal Bid Auction, Lowest Cardinality First

Observe that the winning set $S = \{A_1, \ldots, A_{2\ell+1}\}$ and let $\sigma = (A_1, \ldots, A_{2\ell+1})$ be a permutation of S and $\mathbf{b}^\sigma$ be the resulting ordered maximal bid vector.

Example 5.18. *This is a commodity auction with $2\ell + 2$ agents to purchase $\ell^2 + 1$ identical items. Each agent $A_e \in \{A_1, \ldots, A_{2\ell+2}\}$ has the quantity q_e and cost c_e given in Table 5.18.*

	Agent	q_e	c_e	$b_e^{\min}$	b_e^σ
	A_1	ℓ	0	0	1
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$A_{\ell+1}$	ℓ	0	0	1
	$A_{\ell+2}$	1	0	1	0
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$A_{2\ell+1}$	1	0	0	0
	$A_{2\ell+2}$	$\ell + 1$	1		
Total				1	ℓ

TABLE 5.18: Ordered Maximal Bid Auction, Highest Cardinality First

Observe that the winning set $S = \{A_1, \ldots, A_{2\ell+1}\}$ and let $\sigma = (A_1, \ldots, A_{2\ell+1})$ be a permutation of S and b^σ be the resulting ordered maximal bid vector.

We may also consider choosing the ordering, based on the quantity of items that each agent has. If we were to choose the agents with the largest quantity first, then we could see from Example 5.17 again, that $\frac{b_S^\sigma}{\mathrm{NTUmin}} \geq \Omega(n)$. However, choosing the agents with the smallest quantity first could also result in a ratio of $\frac{b_S^\sigma}{\mathrm{NTUmin}} \geq \Omega(n)$. This can be seen in Example 5.19. Hence, we have seen that the most obvious heuristics for finding a minimal ordering fail for single-commodity auctions, and we leave as an open question whether is is even possible to find a minimal ordering in polynomial time for the special case of single-commodity auctions.

Example 5.19. *This is a commodity auction with $\ell + 3$ agents to purchase $2\ell + 1$ identical items. Each agent $A_e \in \{A_1, \ldots, A_{\ell+3}\}$ has the quantity q_e and cost c_e given in Table 5.19.*

	Agent	q_e	c_e	$b_e^{\min}$	b_e^{σ}
	A_1	1	0	0	1
	A_2	2	0	2	1
	A_3	2	0	0	1
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$A_{\ell+1}$	2	0	0	1
	$A_{\ell+2}$	4	2		
	$A_{\ell+3}$	1	1		
Total				2	$\ell+1$

TABLE 5.19: Ordered Maximal Bid Auction, Lowest Quantity First

Observe that the winning set $S = \{A_1, \dots, A_{\ell+1}\}$ *and let* $\sigma = (A_1, \dots, A_{\ell+1})$ *be a permutation of* S *and* $\mathbf{b}^{\sigma}$ *be the resulting ordered maximal bid vector.*

5.9.4 Results for $|S| \leq 4$

In this subsection, we will see that for any set-system auctions when $|S| \leq 4$ we have OMBmin = NTUmin. This is trivial for $|S| = 1$, and initially we will see this for $|S| = 2$, then $|S| = 3$ before progressing to the main result. Example 5.11 shows that we may have OMBmin>NTUmin for $|S| = 5$, which can easily be applied to all $|S| \geq 5$ (e.g. by substituting agent 1 with k other agents), and so this represents a result for all values of $|S|$. (Intuitively, this threshold of 5 agents is reasonably easy to see using Example 5.11 — it takes at least 3 agents, in an odd-length path, such that their OMB bids are less than the maximum possible. When this triangle bids less than maximum it can, by sharing a constraint, require other agents to bid more than the minimum. Hence at least two of these are needed such that their increases outweigh the decreases made by the agents that are in the triangle.)

As an intermediate step, we will briefly consider a smaller problem, when there are only 2 agents in S.

Lemma 5.8. *Given a set system auction having winning set* S *and* $|S| = 2$, OMBmin = NTUmin.

Proof. We will see that all feasible bid vectors have equal value, and as NTUmin and OMBmin are both feasible bid vectors, it follows that OMBmin = NTUmin.

Assign labels $S = \{A_1, A_2\}$ and define some alternative solutions, as follows.

$$\text{Let } X_1 = \operatorname*{argmin}_{T \in \mathcal{F} \text{ and } A_1 \notin T} c_T,$$

$$\text{let} X_2 = \operatorname*{argmin}_{T \in \mathcal{F} \text{ and } A_2 \notin T} c_T, \text{and}$$

$$\text{let } X_S = \operatorname*{argmin}_{T \in \mathcal{F} \text{ and } S \cap T = \emptyset} c_T.$$

X_1 and X_2 are the cheapest alternative solutions for agents A_1 and A_2 respectively, and are non-empty assuming that the set-system is monopoly-free. X_S is the cheapest alternative solution to both agents in S — and note that there may be no solution to this. Observe that only the sets meeting these definitions could possibly satisfy condition (3) in Definition 1.2 and hence determine the bid values.

Case 1: X_S exists and $c_{X_S \backslash S} \leq c_{X_1 \backslash S} + c_{X_2 \backslash s}$.

As X_1 and X_2 must satisfy condition (2) in Definition 1.2, we have $b_1 \leq c_{X_1 \backslash S}$ and $b_2 \leq c_{X_2 \backslash S}$. However, X_S must also satisfy condition (2), which gives $b_S \leq c_{X_S}$. With the inequality $c_{X_S \backslash S} \leq c_{X_1 \backslash S} + c_{X_2 \backslash S}$, this implies that both X_1 and X_2 cannot simultaneously satisfy condition (3) in Definition 1.2. Hence there must be at least one agent that has condition (3) satisfied by X_S — giving $b_S = c_{X_S}$ for all feasible bid vectors.

Case 2: X_S does not exist or $c_{X_S \backslash S} > c_{X_1 \backslash S} + c_{X_2 \backslash S}$.

If X_S satisfied condition (3) in Definition 1.2, then we would have $b_S > c_{X_1 \backslash S} + c_{X_2 \backslash S}$ hence either $b_1 > c_{X_1 \backslash S}$ or $b_2 > c_{X_2 \backslash S}$ which would violate condition (2) in Definition 1.2. As X_S cannot satisfy condition (3) in Definition 1.2, then both X_1 and X_2 must — giving $b_1 = c_{X_1 \backslash S}$ and $b_2 = c_{X_2 \backslash S}$ for all feasible bid vectors.

□

We now continue by extending this result to winning sets of size 3.

Let $S = \{A_1, A_2, A_3\}$ and label the bids and costs as b_1, b_2, b_3 and c_1, c_2, c_3 respectively.

We will use the existence of the T_e sets (described in (3)) to apply some constraints to the bids depending on the value of $c_{T_e \backslash S}$ for each T_e, in the manner described earlier. Recall that each subset of S may have at most one relevant constraint value. There are seven non-empty subsets of S, and we will label the values for the constraints as $\bar{c}_1, \ldots, \bar{c}_7$. That gives the following possible set of constraints, for any bid vector $\mathbf{b}$.

$$b_1 \leq \bar{c}_1 \tag{5.3}$$
$$b_2 \leq \bar{c}_2 \tag{5.4}$$
$$b_3 \leq \bar{c}_3 \tag{5.5}$$
$$b_1 + b_2 \leq \bar{c}_4 \tag{5.6}$$
$$b_1 + b_3 \leq \bar{c}_5 \tag{5.7}$$
$$b_2 + b_3 \leq \bar{c}_6 \tag{5.8}$$
$$b_1 + b_2 + b_3 \leq \bar{c}_7 \tag{5.9}$$

Taking a NTUmin($\mathbf{c}$) bid vector,$\mathbf{b}^{\min}$, we can examine it in terms of the constraints given here — for each $e \in \{1, 2, 3\}$ at least one of the constraints including b_e must be tight. (in order to satisfy condition (3) in Definition 1.2).

Lemma 5.9. *Given a set system auction having winning set S and $|S| = 3$,* OMBmin = NTUmin.

Proof. We will see that there is an ordering σ such that the maximal ordered bid vector $\mathbf{b}^\sigma$, gives b^σ_S =NTUmin($\mathbf{c}$). There are four possible cases for a set of tight constraints, and we will now examine those in turn.

Case 1: $\exists e \in \{1,2,3\}$, such that $b^\sigma_e = \bar{c}_e$.

For $e \in \{1,2,3\}$, where $\bar{c}_e$ may be tight (i.e., $b^\sigma_e = \bar{c}_e$) then an ordered maximal bid process may firstly raise the bid of e to $b^\sigma_e = \bar{c}_e$. Then we need only consider how to get a minimum bid from the two remaining agents, with just the constraints that apply to them. ($b^\sigma_2 \le \bar{c}_2, b^\sigma_3 \le \bar{c}_3, b^\sigma_2 + b^\sigma_3 \le \min(\bar{c}_6, \bar{c}_7 - b^\sigma_1)$). Lemma 5.8 tells us that either ordering will give us a minimum bid, and hence b^σ_S =NTUmin($\mathbf{c}$).

Case 2: $b^\sigma_1 + b^\sigma_2 + b^\sigma_3 = \bar{c}_7$ holds.

Where $b^\sigma_1 + b^\sigma_2 + b^\sigma_3 = \bar{c}_7$ holds in a NTUmin($\mathbf{c}$) bid vector, then we have NTUmin($\mathbf{c}$) $= \bar{c}_7$. As the constraint $b^\sigma_1 + b^\sigma_2 + b^\sigma_3 \le \bar{c}_7$ applies to all bid vectors that satisfy (2), then any such bid vector has $b^\sigma_1 + b^\sigma_2 + b^\sigma_3$ =NTUmin($\mathbf{c}$). Observe that the ordered maximal bid process must give a bid vector that satisfies (1), (2) and (3), hence when $b^\sigma_1 + b^\sigma_2 + b^\sigma_3 \le \bar{c}_7$ may be tight the ordered maximal bid process gives b^σ_S =NTUmin($\mathbf{c}$).

Case 3: Exactly two of $\{b^\sigma_1 + b^\sigma_2 = \bar{c}_4, b^\sigma_1 + b^\sigma_3 = \bar{c}_5, b^\sigma_2 + b^\sigma_3 = \bar{c}_6\}$ hold.

Observe that when only 1 may be tight, either Case 1 or Case 2 above must apply, otherwise there is some $e \in S$ that does not have a tight constraint.

Then we can choose an e that occurs most frequently to raise first. We will assume w.l.o.g. that the two tight constraints are as follows;

$$b^\sigma_1 + b^\sigma_2 = \bar{c}_4$$
$$b^\sigma_2 + b^\sigma_3 = \bar{c}_6$$

and we can observe that $b^\sigma_S = b^\sigma_1 + b^\sigma_2 + b^\sigma_3$ and $b^\sigma_1 + b^\sigma_2 + b^\sigma_3 = \bar{c}_4 + \bar{c}_6 - b^\sigma_2$. Hence when b^σ_2 is maximized, b^σ_S is minimized, which is proven as follows.

Assume, for contradiction, that there exists a bid vector $\mathbf{b}'$ where $b'_2 = b^\sigma_2 - \delta$ for some $\delta > 0$ and that $b'_S < b^\sigma_S$. As both $b'_1 + b'_2 = \bar{c}_4$ and $b^\sigma_1 + b^\sigma_2 = \bar{c}_4$ will hold, we can rearrange to give $b'_1 = b^\sigma_1 + \delta$. The same applies to $b^\sigma_2 + b^\sigma_3 = \bar{c}_6$, giving $b'_3 = b^\sigma_3 + \delta$. Therefore $b'_S = b^\sigma_S - \delta + \delta + \delta$, contradicting $b'_S < b^\sigma_S$.

In order to maximize b^σ_2 we simply place A_2 first in the ordering. We can then consider only agents 1 and 3, and the constraints that apply to them, and Lemma 5.8 shows us that we can minimize these bids with an ordered maximal bid.

Case 4: $b^\sigma_1 + b^\sigma_2 = \bar{c}_4$, $b^\sigma_1 + b^\sigma_3 = \bar{c}_5$ and $b^\sigma_2 + b^\sigma_3 = \bar{c}_6$ hold.

This can be rewritten to give $2(b_1^\sigma+b_2^\sigma+b_3^\sigma) = \bar{c}_4+\bar{c}_5+\bar{c}_6$, hence NTUmin($\mathbf{c}$) $= 1/2(\bar{c}_4+\bar{c}_5+\bar{c}_6)$. Reverting back to the inequality form of these equations, they represent upper bounds on any bid that satisfies condition (2) in Definition 1.2. Therefore we have for any such vector b′, $2(b_1'^\sigma + b_2'^\sigma + b_3'^\sigma) \leq \bar{c}_4 + \bar{c}_5 + \bar{c}_6$. As an ordered maximal bid vector $\mathbf{b}^\sigma$, for any possible order σ, satisfies condition (2) then this gives $2b_S^\sigma \leq \bar{c}_4 + \bar{c}_5 + \bar{c}_6$ and therefore $b_S^\sigma \leq$NTUmin($\mathbf{c}$). As NTUmin($\mathbf{c}$) is a minimum, it follows that $b_S^\sigma \geq$NTUmin($\mathbf{c}$).

Case 1 deals with any case where $b_e^\sigma = \bar{c}_e$, and Case 2 where $b_1^\sigma+b_2^\sigma+b_3^\sigma = \bar{c}_7$. Hence any other cases left must only have some subset of $\{b_1^\sigma + b_2^\sigma = \bar{c}_4, b_1^\sigma + b_3^\sigma = \bar{c}_5, b_2^\sigma + b_3^\sigma = \bar{c}_6\}$. In order for all agents to have some tight constraint, at least two of these must be tight, and hence Case 3 and Case 4 cover these. Therefore all the possible sets of tight constraints must be covered by at least one of the 4 cases already outlined.

As we have now seen that all possible cases have an ordering σ which will give the value b_S^σ =NTUmin($\mathbf{c}$), the claim that there always exists an ordering σ giving b_S^σ =NTUmin($\mathbf{c}$) is proven. □

Now we will consider when $|S| = 4$; firstly we note (in the same way as before) that we can use the fact that there is a 'locally optimal' solution for any three of the agents.

Theorem 5.10. *Given a set-system auction, having the winning set S with $|S| \leq 4$ then* OMBmin = NTUmin.

Proof. Let $\mathbf{b}^{\min}$ be a NTUmin bid vector.

When $|S| = 4$, there are a small number of configurations for the tight constraints. Firstly, we can assume that the tight constraints make a single connected component. If not, then the components can be ordered separately, and each component (of size ≤ 3) has an ordering that gives a minimum value, from Lemma 5.8 and Lemma 5.9.

We saw in Lemma 5.8 that when we have only 2 agents, then either ordering gives the same value, hence the minimum and maximum are equal. We can now consider the different ways that a set of size 4 can have tight constraints.

Case 1: All tight constraints are of size 2.

Firstly; we will consider only those that do not have a tight constraint of size 3; hence all the tight constraints are of size 2. We will do this as sub-cases.

Case 1.1: There is no odd-length cycle of tight constraints.

When there is no odd-length cycle (i.e., no triangle) then the resulting tight constraints can be represented as a bipartite graph. W.l.o.g. assume that we have partitions $S_1 = \{A_1, A_3\}, S_2 = \{A_2, A_4\}$ such that these two partitions are bipartite; i.e., $\forall k \in \{1,2\}, i \in A_k \Leftrightarrow N(i) \notin A_k$. We can observe that increasing the bid of some agent in S_1 would then reduce the bid of at least one agent in S_2 by the same amount (as they share a tight constraint). When increasing the bid of one agent in S_1 would reduce the bids of both agents in S_2 (this must be by the same amount), then the other agent in S_1 will be sharing a tight constraint with one agent in S_2, and hence it will be able to increase its bid by the same amount. This shows that increasing a bid to its maximum possible value will have no effect on the overall bid. Therefore, by taking a NTUmin bid, it would be trivial to assign any ordering to the bids which may 'amend' the individual bids as described, but would still produce an equal sum, and hence NTUmin($\mathbf{c}$).

Case 1.2: There is an odd-length cycle of tight constraints.

Where there exists an odd-length cycle of tight constraints, then we have a triangle of tight constraints, along with one other tight constraint attached to one of the nodes of the triangle. Hence, there is some agent (A_1) that shares a tight constraint with 3 neighbours. Therefore, any minimum bid vector must maximize $b_1^{\min}$. Assume, for contradiction, that $\mathbf{b}^{\min}$ is a minimum bid vector, but $b_1^{\min}$ is not maximized — hence by the tight constraints, there exists some $\mathbf{b}'$ when $b'_1 = b_1^{\min}+\epsilon$ then $b'_2 = b_2^{\min} - \epsilon$, $b'_3 = b_3^{\min} - \epsilon$ and $b'_4 = b_4^{\min} - \epsilon$, giving $b'_S < b_S^{\min}$ contradicting $\mathbf{b}^{\min}$ being a minimum. As $b_1^{\min}$ may be maximized by choosing it first in any ordering, there exists an ordered maximal bid vector $\mathbf{b}^{\sigma}$ when b_1^{σ} is maximum. When b_1^{σ} is maximum, then the bids of all other agents can be raised in any order, as each shares a tight constraint with A_1; thus showing that when b_1^{σ} is maximized, then b_S^{σ} is minimized, and we have b_S^{σ} =NTUmin($\mathbf{c}$).

Case 2: We have some tight constraint of size 3.

We will also examine this as sub-cases.

Case 2.1: There are two sets of tight constraints, both of size 3.

There are two agents that exist in both these sets of tight constraints. Ordering these two agents first, in either order, will result in a maximum for the sum of their bids. (This follows from Lemma 5.8. When these two agents have raised their bids, the other two agents can raise theirs, in either order, to make the constraints tight.

Case 2.2: There are at least three sets of tight constraints of size 3.

As there is at least one agent that is in all sets of tight constraints, then clearly the bid for this agent must be maximized. By ordering this agent first, its bid is maximized and hence the other three agents can be raised in order to make the constraints tight.

Case 2.3: There is one set of tight constraints of size 3.

Case 2.3.1: There is one set of tight constraints of size 2 and one set of tight constraints of size 3.

In order for all agents to be involved in a tight constraint, there is at least one constraint of size 2. When there is only one constraint of size 2, there is one agent that appears in both sets of tight constraints, let us assume that this is agent A_1.

Let $\bar{c}_1$ and $\bar{c}_2$ be the values of the constraints, and the sum of the bids will equal $\bar{c}_1 + \bar{c}_2 - b_1^\sigma$. Hence if we order this agent first, its bid is maximized and the sum of bids is minimized; then the other three agents can be raised in order to make the constraints tight.

Case 2.3.2: There are two sets of tight constraints of size 2 and one set of tight constraints of size 3.
When there are two constraint of size 2, there are three agents that appear in these two sets of tight constraints, let us assume that they are agents A_1,A_2,A_3. Let $\bar{c}_1$, $\bar{c}_2$ and $\bar{c}_3$ be the values of the constraints, and the sum of the bids will equal $\bar{c}_1 + \bar{c}_2 + \bar{c}_3 - b_1^\sigma - b_2^\sigma - b_3^\sigma$. Hence if we order agent A_4 last, its bid is minimized and the sum of bids is minimized.

Case 2.3.3: There are three sets of tight constraints of size 2 and one set of tight constraints of size 3.
As all agents are covered by tight constraints of size 2, case 1 above applies, and shows that an optimal ordering σ gives b_S^σ =NTUmin($\mathbf{c}$).

$\square$

5.9.5 Upper Bound for Restricted Settings

We will now return to the special case described in Section 5.9.3. Recall that, in this setting every constraint is restricted to a single value (we assume this to be 1), the cost of the winning set is $c_S = 0$ and that the constraints are binary — every constraining set $(T \setminus S)$ has a cardinality of 2.

Recall Example 5.14, where we saw that even in this restricted setting, NTUmin($\mathbf{c}$) may not be achievable with a maximal ordered bid vector. However, our examples have only shown a ratio of OMBmin($\mathbf{c}$)/NTUmin($\mathbf{c}$) ≤ 2, while for ternary constraints in Example 5.13 we saw ratios larger than any constant OMBmin($\mathbf{c}$)/NTUmin($\mathbf{c}$) $= 2n/9$.

So, we may ask the question, what is the largest ratio possible between the OMBmin and NTUmin($\mathbf{c}$) in this restricted setting? We will see that OMBmin($\mathbf{c}$) is always within a factor of 2 of NTUmin($\mathbf{c}$).

We will firstly show how to convert a NTUmin bid vector $\mathbf{b}^{\min}$, into another bid vector $\mathbf{b}$, where $b_S = b_S^{\min}$, and $\mathbf{b}$ satisfies some additional properties on the bid values. We will then use these properties to compare $\mathbf{b}$ with an OMBmin bid vector $\mathbf{b}^\sigma$.

Lemma 5.11. *For an instance of a set-system auction with binary single-value constraints, a given winning set S such that $c_S = 0$ and a given* NTUmin *bid vector* $\mathbf{b}^{\min}$, *a bid vector* $\mathbf{b}$ *exists such that* $b_S^{\min} = b_S$ *and* $\forall e \in S, b_e \in \{0, 1/2, 1\}$.

Proof. Consider a NTUmin bid vector $\mathbf{b}^{\min}$, we will show how to create some bid vector, $\mathbf{b}$ such that $b_S^{\min} = b_S$ and $\forall e \in S, b_e \in \{0, 1/2, 1\}$.

Let $d \in [1/2, 1)$ be a bid value, and partition S into subsets for each distinct value of d.

$$\text{Let } S_{d+} = \bigcup_{e \in S, b_e^{\min} = d} \{e\}.$$

$$\text{Let } S_{d-} = \bigcup_{e \in S, b_e^{\min} = 1-d} \{e\}.$$

(i.e., if $d = b_e \in [1/2, 1)$ then $e \in S_{d+}$ and if $d = b_e \notin [1/2, 1)$ then $e \in S_{d-}$). Now fix d as some bid value, such that $S_{d+} \neq \emptyset$. We can observe that every $e \in S_{d-}$ has a neighbour $j \in N(e)$ such that $j \in S_{d+}$ (If no such neighbour exists, then there is no tight constraint that would give $b_e^{\min} + b_j^{\min} = 1$ contradicting e being in S_{d-}). Similarly every $e \in S_{d+}$ has a neighbour $j \in N(e)$ such that $j \in S_{d-}$. When $b_S^{\min}$ is optimal, this implies that $|S_{d+}| = |S_{d-}|$, as follows;

Assume, for contradiction, that $|S_{d+}| \neq |S_{d-}|$ and $b_S^{\min}$ is a minimum. If $|S_{d+}| > |S_{d-}|$, then choosing some small ϵ, there exists a bid vector $\mathbf{b}$ where $e \in S_{d+} \Leftrightarrow b_e = b_e^{\min} - \epsilon$ and $e \in S_{d-} \Leftrightarrow b_e = b_e^{\min} + \epsilon$. For all $e \notin (S_{d+} \cup S_{d-})$ let $b_e = b_e^{\min}$. The only agents that share a tight constraint with agents in S_{d_+} must be present in S_{d-}, and vice-versa. Let ϵ be small enough such that no other constraint may become tight in $\mathbf{b}$ that was not tight in $\mathbf{b}^{\min}$. As this would now allow more agents to decrease by ϵ than increase by ϵ this contradicts $b_S^{\min}$ being a minimum. Similarly, if $|S_{d-}| > |S_{d+}|$ there exists a bid vector $\mathbf{b}'$ where $e \in S_{d+} \Leftrightarrow b'_e = b_e^{\min} + \epsilon$ and $e \in S_{d-} \rightarrow b'_e = b_e^{\min} - \epsilon$. This would also allow more agents to decrease by ϵ than increase by ϵ, which contradicts $b_S^{\min}$ being a minimum.

We have seen that each agent $e \in S_{d+}$, there exists at least one neighbour $j \in N(e)$ such that $b_e^{\min} + b_j^{\min} = 1$, and hence that $b_j^{\min} = 1 - b_e^{\min}$. For each d, when $S_{d+} \neq \emptyset$, for each $e \in S_{d+}$ let $b_e = 1$, and for each $e \in S_{d-}$ let $b_e = 0$. As, in each turn, we have added $1 - d$ to the bid of elements in S_{d+} and subtracted $1 - d$ from the bid of elements in S_{d_-}, we will have $b_{S_{d+}} = b_{S_{d+}}^{\min} + |S_{d_+}|d$ and $b_{S_{d-}} = b_{S_{d-}}^{\min} - |S_{d_-}|d$. As we have seen that $|S_{d_+}| = |S_{d_-}|$, this gives $b_{S_{d+} \cup S_{d-}} = b_{S_{d+} \cup S_{d-}}^{\min}$ and, summing over all d, we have $b_S = b_S^{\min}$.

This has shown that $\mathbf{b}$ exists such that $\forall e \in S, b_e \in \{0, 1/2, 1\}$ and that $b_S = b_S^{\min}$, therefore there exists some bid vector $\mathbf{b}$ such that $b_S =$NTUmin($\mathbf{c}$) and $\forall e \in S, b_e \in \{0, 1/2, 1\}$ which satisfies the lemma.

□

Now that we have seen the existence of a NTUmin bid vector $\mathbf{b'}^{\min}$ with this property ($\forall e \in S, b_e \in \{0, 1/2, 1\}$) on the bid values, we will see how an ordering σ can be derived such that $b_S^\sigma \leq 2b_S'^{\min}$.

Theorem 5.12. *Let I be an an instance of a set-system auction having binary single-value constraints and a winning set S such that $c_S = 0$. Then there exists an ordering σ such that the resulting bid vector $\mathbf{b}^\sigma$ has $b_S^\sigma \leq 2$NTUmin($\mathbf{c}$).*

Proof. Let $\mathbf{b}^{\min}$ be a NTUmin bid vector, satisfying the property that for all $e \in S, b_e^{\min} \in \{0, 1/2, 1\}$. Lemma 5.11 has shown that such a bid vector exists for any instance I of a set-system auction with binary single-value constraints.

Let σ be an ordering of S in decreasing order, according to an agents bid in $\mathbf{b}^{\min}$. More formally, σ is an ordering of S such that $\forall (i,j) \in S, \sigma_i < \sigma_j \Leftrightarrow b_i^{\min} \geq b_j^{\min}$. (That is, if an agent i appears before agent j in the ordering σ, then agent i did not bid less than agent j in the NTUmin bid vector $\mathbf{b}^{\min}$). Partition S into $S_0, S_{1/2}, S_1$, based on the bids in $\mathbf{b}^{\min}$, i.e., $\forall i \in S_x, b_i^{\min} = x$.

Now we claim that every agent $i \in S_1$ will bid $b_i^\sigma = 1$. For any agent i when $b_i^{\min} = 1$ there is no neighbour $j \in N(i)$ such that $b_j^{\min} > 0$ or $\mathbf{b}^{\min}$ would violate a shared constraint (recall that being neighbours implies $b_i^{\min} + b_j^{\min} \leq 1$). As the ordering σ will require that i is raised before all of its neighbours, then we will have $b_i^\sigma = 1$ and $\forall j \in N(i), b_j^\sigma = 0$. Equally, to show that $\forall i \in S_0$ then $b_i^\sigma = 0$, then i must have some neighbour $j \in N(i)$ that has $b_j^{\min} = 1$, (or else i has no tight constraint). As, we have seen that $b_j^\sigma = 1$ will already be set when we reach agent i in σ, then we must have $b_i^\sigma = 0$ or else we would have $b_i^\sigma + b_j^\sigma > 1$, violating the shared constraint.

It is trivial to observe that no agent $i \in S$ may bid more than 1, as every agent has some constraint, so it follows that $\forall i \in S_{1/2}, b_i^{\min} \leq 1$. We can write the sum of bid vector $\mathbf{b}^{\min}$, as follows

$$b_S^{\min} = |S_1| + (1/2)|S_{1/2}|$$

and similarly we can upper bound the bid vector b_S^σ using the results we have just seen on individual bids.

$$b_S^\sigma \leq |S_1| + 2|S_{1/2}|$$

rewrite to give an upper bound

$$b_S^\sigma \leq 2(|S_1| + |S_{1/2}|)$$

and as $b_S^{\min} = |S_1| + (1/2)|S_{1/2}|$ then $b_S^{\min} \leq |S_1| + |S_{1/2}|$ and by substitution, we have

$$b_S^\sigma \leq 2(b_S^{\min})$$

As we have assumed that $b_S^{\min} = \text{NTUmin}(\mathbf{c})$, this shows that for the ordering σ defined earlier, we will get an ordered maximal bid vector $\mathbf{b}^\sigma$ such that $b_S^\sigma \leq 2\ \text{NTUmin}(\mathbf{c})$, as claimed.

□

5.9.6 Hardness and Approximation Results

We will firstly see a straightforward reduction from the MINIMUM INDEPENDENT DOMINATING SET problem on a graph G, to finding OMBmin($\mathbf{c}$) on a set system derived from G. We will use this fact to show that finding OMBmin($\mathbf{c}$) is NP-hard to even approximate.

We now give the definition of a known problem, Problem 5.

Name MINIMUM INDEPENDENT DOMINATING SET (MIN-IDS)

Instance Graph $G = (V, E)$.

Output Cardinality of a *minimum independent dominating set* for G, i.e., $|V'|$ for a subset $V' \subseteq V$ such that for all $u \in V - V'$ there is a $v \in V'$ for which $(u, v) \in E$, and such that no two vertices in V' are joined by an edge in E.

PROBLEM 5: MINIMUM INDEPENDENT DOMINATING SET

The decision version of Problem 5 is well-known to be NP-complete [14].

Lemma 5.13. *For any graph G, there exists a polynomial-time reduction to a set-system auction with cost vector* $\mathbf{c}$ *such that* OMBmin($\mathbf{c}$) *equals the size of a minimum independent dominating set of G.*

Proof. Taking a graph $G = (V, E)$, create a set-system auction I as follows.

Copy the vertices with cost 0.

$$\mathcal{E} = \{A_1, \ldots, A_n\} = \{V_1, \ldots, V_n\}$$

$$c_1, \ldots, c_n = 0$$

Create a new agent, A_0, with cost 1.

$$\mathcal{E} = \mathcal{E} \cup \{A_0\}$$

$$c_0 = 1$$

Note the winning set S.

$$S = \{A_1, \ldots, A_n\}$$

Create feasible sets for every edge $e \in E$, with endpoints V_i, V_j

$$\forall e \in \mathcal{E}, T_e = S \setminus \{A_i, A_j\} \cup A_0$$

Define the feasible sets.

$$\mathcal{F} = S \cup \bigcup_{e \in E} T_e$$

Observe that there are at most $n + 1$ agents in I, hence I can be created from G in polynomial time. Let $\mathbf{b}^\sigma$ be the Ordered Maximal bid vector, for the optimal ordering σ (i.e., $\mathbf{b}^\sigma_S$ =OMBmin), and let M be a minimum independent dominating set of G.

There exists a constraint, on each edge $e \in E$ such that agents at the two end points V_i and V_j may not bid such that $b^\sigma_i + b^\sigma_j > 1$ (or else feasible set T_e would give $b^\sigma_{S \setminus T_e} > c_{T_e \setminus S}$ showing that $\mathbf{b}^\sigma$ does not satisfy condition (2) in Definition 1.2).

Observe that $\forall e \in S, b_e^\sigma \in \{0,1\}$, as each bid starts at 0, and then takes the form $b_e^\sigma = 1 - \max_{j \in N(e)} b_j^\sigma$. We partition the agents into S_0 and S_1, depending on their bid, as follows. Let $S_x \subseteq S$ be a set such that $\forall e \in S_x, b_e^\sigma = x$. Recall that $N(e)$ is the set of neighbours of agent e; the constraint property of condition (2) in Definition 1.2 implies that $\forall e \in S_1, N(e) \notin S_1$ — i.e., that no two neighbours may bid one. This implies that S_1 is an independent set in G. The 'tightness' property of condition (3) in Definition 1.2 requires that $\forall e \in S, (\{e\} \cup N(e)) \cap S_1 \neq \emptyset$ — if an agent bids zero, one of its neighbours must bid one. This implies that S_1 is a dominating set in G.

Therefore, the set S_1 is an independent dominating set in G, and is at least as large as M, by definition. As OMBmin(**c**) $= |S_1|$, from the definition of S_0 and S_1, this shows that OMBmin $\geq |M|$.

For the other direction, let σ be an ordering such that every agent in M appears before all agents not in M. As M is an independent set, each agent in M can raise its bid to 1 while its neighbours still bid 0. As M is dominating, no agent outside M will then be able to bid more than 0 due to the presence of a neighbour in M already bidding 1. Hence $b_S^\sigma = |M|$, and OMBmin $\leq |M|$.

This shows that computing OMBmin(**c**) gives the size of the minimum independent dominating set.

□

As we have seen a proof that computing OMBmin can be used to compute the size of a minimum independent dominating set of a graph, this is sufficient for an approximation hardness result.

Theorem 5.14. *For any set-system auction I, having cost vector* **c**, *there is no constant $\epsilon > 0$ for which* OMBmin(**c**) *can be approximated within a factor of $n^{1-\epsilon}$ in polynomial time, unless* P=NP.

Proof. From Lemma 5.13 we see that for any graph, G, the size of the minimum independent dominating set can be calculated by determining OMBmin(**c**) for instance I (which is derived from G in polynomial time). Therefore, if OMBmin(**c**) could be approximated within $n^{1-\epsilon}$ for instance I, this would imply a polynomial-time approximation within a factor of $n^{1-\epsilon}$ for the minimum independent dominating set of G, and [19] shows that this would imply P=NP. □

We will firstly see a technical theorem, that shows for particular (bipartite) restricted set systems, that there exists a NTUmin bid vector such that all agents bid either 0 or 1. We will use this fact to show that, for these cases, NTUmin(**c**) is equal to OMBmin(**c**). We can then leverage the result of Theorem 5.14 to apply to computing NTUmin(**c**), showing that NTUmin(**c**) is similarly hard to approximate.

Considering the hypergraph representation of set-system constraints described in Section 5.2, we are interested in those set-systems that result in a bipartite graph. Recall that each constraint imposed by the set-system must have cardinality 2, (i.e.,

$\forall T \in \mathcal{F} \setminus \{S\}, |S \setminus T| = 2$) which can be represented by an edge in a graph. We will be interested in auction instances such that this resulting constraint graph is bipartite, and will refer to these as 'bipartite' constraint sets.

Theorem 5.15. *For any set-system auction I, having cost vector* $\mathbf{c}$*, there is no constant $\epsilon > 0$ for which* NTUmin($\mathbf{c}$) *can be approximated within a factor of $n^{1-\epsilon}$ in polynomial time, unless* P=NP.

Proof. We will firstly show that, for any instance I with binary and bipartite constraint sets, a given winning set S such that $c_S = 0$ and a given NTUmin bid vector $\mathbf{b}^{\min}$ then a vector $\mathbf{b}$ exists such that $b_S = b_S^{\min}$ =NTUmin and $\forall e \in S, b_e^{\sigma} \in \{0, 1\}$.

Let $\mathbf{b}^{\min}$ be NTUmin bid vector $\mathbf{b}^{\min}$ such that $\forall e \in S, b_e^{\min} \in \{0, 1/2, 1\}$, Lemma 5.11 shows that such a bid vector exists. We know that every agent e with $b_e^{\min} = 1/2$ has at least one neighbour, y with $b_y^{\min} = 1/2$. Let S_h be a subset of S such that $\forall e \in S_h, b_e^{\min} = 1/2$.

For all $e \in S \setminus S_h$ let $b_e = b_e^{\min}$.

Divide S_h into two partitions, S_{h1} and S_{h2}, such that $\forall e \in S_{h1}$, there does not exist a y such that $(y \in (N(e)))$ and $(y \in S_{h1})$ (there are no edges between vertices that are in the same partition). This is possible only because we require the constraint set to be bipartite.

No agent in S_h has a neighbour outside S_h with a non-zero bid; as any such agent e would have a neighbour j with $b_j^{\min} = 1$ and hence $b_e^{\min} + b_j^{\min} > 1$ violating their shared constraint. As all agents in S_{h1} have neighbours in S_{h2} and vice-versa (or else we do not have any e, j such that $b_e^{\min} + b_j^{\min} = 1$), then we can choose new bid values of 0 and 1.

More formally we have $e \in S_{h1}$ if and only if $b_e = 1$ and $e \in S_{h2}$ if and only if $b_e = 0$. If $|S_{h1}| = |S_{h2}|$ then we have $b_{S_h}^{\min} = b_{S_h}$. Clearly we may not have $|S_{h1}| < |S_{h2}|$ as this would give $b_S < b_S^{\min}$ contradicting $b_S^{\min}$ being a minimum vector. Simply swapping the labels for sets S_{h1} and S_{h2} would equally imply that $|S_{h1}| > |S_{h2}|$ is not possible.

As this shows we have $b_{S_h}^{\min} = b_{S_h}$, and we have assigned bids such that $\forall e \in S, b_e \in \{0, 1\}$, it implies that any such restricted system with bipartite constraint sets has a NTUmin bid vector with $\forall e \in S, b_e \in \{0, 1\}$. We can observe that any ordering σ gives an ordered maximal bid vector $\mathbf{b}^{\sigma}$ such that $\forall e \in S, b_e^{\sigma} \in \{0, 1\}$. Let σ be an ordering such that all agents with a bid bid of 1 in $\mathbf{b}$ appear before any agent that bids 0, which gives an ordered maximal bid vector of $\mathbf{b}^{\sigma} = \mathbf{b}$. Therefore, when the set system has binary and bipartite constraint sets, we have NTUmin($\mathbf{c}$)= b_S^{σ}, for a minimum ordering σ, therefore NTUmin($\mathbf{c}$)=OMBmin($\mathbf{c}$).

If we take a bipartite input graph, G, then from Lemma 5.13 this would give the size of the minimum independent dominating set on G. Similarly to Theorem 5.14, if we could approximate NTUmin($\mathbf{c}$) within $n^{1-\epsilon}$ for instance I, this would imply a polynomial-time approximation within $n^{1-\epsilon}$ for the minimum independent dominating set of the bipartite graph G, and [7], shows that this would imply P=NP.

□

As finding NTUmin (or OMBmin) involves firstly finding the lowest-cost solution to a possibly inapproximable problem, an approximation hardness result may appear to be trivial. However, it is also easy to see that these approximation hardness results for NTUmin and OMBmin can be applied even when the underlying problem is polynomial-time computable. One such example is a problem based on MINIMUM WEIGHT EDGE COVER.

Name MINIMUM WEIGHT EDGE COVER

Instance A graph $G = V, E$ and an edge-weight function $w(e)$ for $e \in E$).

Output A set of edges $S \subseteq E$ such that every vertex $v \in V$ is incident to an edge e in S and $\sum_{e \in S} w(e)$ is minimized.

PROBLEM 6: MINIMUM WEIGHT EDGE COVER

MINIMUM WEIGHT EDGE COVER is known to be polynomial-time solvable [32].

It was shown in [11] that NTUmin(c) is NP-hard to calculate exactly, even when the problem of finding the minimum-cost solution is polynomial-time solvable.

Given an arbitrary bipartite input graph $G = (V, E)$, assuming at least one edge in E, define an instance of WEIGHTED EDGE COVER as follows.

$$\text{Let } V' = V \cup \{A_0\}.$$

$$\text{let } E' = E \cup \bigcup_{v \in V} \{\{v, A_0\}\}.$$

$$\text{let } w'(e) = \begin{cases} 0, & \text{if } A_0 \in e \\ 1, & \text{otherwise.} \end{cases}$$

From this instance of MINIMUM WEIGHT EDGE COVER, we create a set-system auction, with the feasible sets as the valid solutions to the edge cover problem, as follows. Let $\mathcal{E} = E'$, Let $\mathcal{F} = \{T : \forall v \in V', \exists D \in T \text{ such that } v \in D\}$, and let $c_e = w'(e)$ for all $e \in \mathcal{E}$.

Observe that the set $S = \bigcup_{v \in V} \{\{v, A_0\}\}$ covers all vertices in V' and has weight 0 (and hence cost 0), so may be chosen as a winning set.

We define a subset of the feasible sets that differ from the winning set by only one 'edge'.

$$\text{Let } \mathcal{F}_1 = \left\{ S, \bigcup_{(u,v) \in E} S \setminus \{\{u, A_0\}, \{v, A_0\}\} \cup \{u, v\} \right\}.$$

Observe that $\mathcal{F}_1 \subseteq \mathcal{F}$ (every feasible set $T \in \mathcal{F}_1$ has $w(T) \leq 1$, and every feasible set $T' \notin \mathcal{F}_1$ has $w(T') > 1$).

Consider that every feasible set $T \in (\mathcal{F}_1 \setminus \{S\})$ gives $S \setminus T = \{u, v\}$ for some $(u, v) \in E$ and $c_{T \setminus S} = 1$. This shows that the constraint implied by condition (2) in Definition 1.2 on any NTUmin bid vector $\mathbf{b}^{\min}$ may be given by $\mathbf{b}_u^{\min} + \mathbf{b}_v^{\min} \leq 1$, hence we can observe that the constraints for this set-system mirror the input graph G. (We can ignore the constraints with values greater than 1, as they would imply some agent bidding more than 1, which would already violate an 'edge' constraint). NTUmin must allocate bids such that every two neighbours $(u, v) \in E$ must bid $b_u^{\min} + b_v^{\min} \leq 1$ (from condition (2)) and that $\forall u \in V, \exists (u, v) \in E$ such that $b_u^{\min} + b_v^{\min} = 1$ (from condition (3) in Definition 1.2), and hence NTUmin gives exactly the size of a minimum independent dominating set in graph G. Hence, there are auctions based on instances of the polynomial-time solvable problem MINIMUM WEIGHT EDGE COVER where it is hard to even approximate NTUmin.

Chapter 6

Benchmarks for Forward Auctions

6.1 Overview

Our area of study has been, thus far, that of procurement auctions, where some central authority is distributing revenue in return for some service. In this chapter we take the natural step of considering the opposite approach, where the central authority may be distributing services in return for revenue.

The set-system auctions of Karlin et al. [24] that we have been studying are often referred to as an auction for 'hiring a team' (e.g. [4, 36, 24, 11]); they nicely characterize the concept. We may like to consider a forward set-system auction as one of 'providing a service' and we give a definition of such an auction here. In these auctions we assume that we have a single seller that is able to provide services, and that there are various buyers that would wish to purchase these services — but that it may only be possible for certain subsets of buyers to receive these services simultaneously. As the seller must choose which subsets he will provide to, we will consider that those who receive a service as 'winners', and those that do not as 'losers'.

As we have been interested in frugality for set-system auctions, we would like to consider the same concept for these 'providing a service' auctions. In order to do that we will need to consider what we might use as a benchmark figure, a question which this chapter aims to go some way to addressing.

We will firstly see a definition for a forward set-system auction, and show some comparison with the more commonly studied combinatorial auction. We then examine some possibilities for computing a reasonable benchmark figure, and compare it with the benchmarks that are already used for the unit-demand special-case of the combinatorial auction.

6.2 Definitions

In these auctions we assume that the seller can perform some service and that there are also 'feasible sets' of buyers who could be served.

We define a forward set-system auction analogously to the set-system auction described in Section 1.4.1.

Let a set system $(\mathcal{E}, \mathcal{F})$ be specified by a set $\mathcal{E}$ of n elements, each representing an agent, and a collection $\mathcal{F} \subseteq 2^{\mathcal{E}}$ of feasible sets; these are the subsets of agents that make a possible solution for the seller (i.e., for all $T \in \mathcal{F}$ then every agent $e \in T$ can be served simultaneously).

Let the valuation vector $\mathbf{v} = (v_1, \ldots, v_n)$ represents the (private) valuation v_e that each agent e will place on being chosen at the auction. For ease of notation, let $v_W = \sum_{e \in W} v_e$ be an aggregate function.

Each buyer e then makes a bid b_e to the mechanism. The mechanism may then choose a 'winning' set S of agents to serve, and ask that each agent $e \in S$ will pay some price $p_e \in [0, v_e]$ for the service provided and agents that are not in the winning set do not pay, $\forall e \notin S, p_e = 0$. These are the commonly-made assumptions of 'no positive transfers' and 'voluntary participation' (see, e.g., [35]).

Unlike some other forms of auction, we only consider that agents are 'winning' (in the chosen solution) or 'losing' (not in the chosen solution). We do not consider that an agent could receive more than one service or that the agent may value the services differently.

Raising revenue by auction has often been studied in the literature, as *Combinatorial Auctions* (e.g. [38, 35]). We now consider the similarities between these combinatorial auctions and the set-system auctions proposed.

6.3 Comparison with Combinatorial Auctions

In order to provide some context for a comparison, we begin with a commonly-used description of a Combinatorial Auction.

6.3.1 Definitions

There is a set of m indivisible items and a set of n bidders that each wish to purchase some combination(s) of these items. Every bidder e has a valuation function v_e, which specifies the value that bidder e has for each subset (or bundle) of items. We assume that these valuations are monotonic — that $S \subseteq T \rightarrow v_e(S) \leq v_e(T)$ and that $v_e(\emptyset) = 0$.

A solution to a combinatorial auction is an allocation of the items $(A_1, \ldots, A_n)$ to the bidders, when A_e is the set of items allocated to agent e, such that no item is allocated to more than one bidder. Each bidder pays some price p_e, depending on the allocation the bidder receives. (We assume that if $v_e(A_e) = 0$ then $p_e = 0$, that any agent does not pay if he does not receive a bundle that he was interested in). Let $\mathcal{A}$ be the set of all possible allocations.

There are different ways of looking at the effectiveness of Combinatorial Auctions, such as maximizing the social welfare (i.e., $\sum_{e \in \{1,\ldots,n\}} v_e(A_e)$), but as our interest is with the payment bounds, we will be concerned with the total revenue raised by the auction (i.e., $\sum_{e\in\{1,\ldots,n\}} p_e$).

6.3.2 Comparison of Set-System Auctions and Combinatorial Auctions

Both standard combinatorial auctions and our forward set-system auctions deal with the same scenario — when bids are invited, by some authority, for the purchase of goods or services. We will briefly look at some of the differences between the two settings.

Very often, combinatorial auctions are considered with *free-disposal* — that not all items must be allocated. This is also generally assumed in the case of reverse set-system auctions, as there is never any benefit in making payments to more agents than are strictly necessary.

The value of this assumption is not so obvious for a forward set-system auction, as it may actually raise more revenue by choosing to sell to fewer agents, so the difference between the two settings (of having open or closed sets) may be pertinent. For example, consider that a truthful mechanism wishes to sell two items to more than three buyers. However, only two of the buyers have a non-zero valuation for the items. A truthful mechanism could sell two items, at a price of zero, or possibly actually just sell one item at the second-highest price, and gain some revenue. Therefore, it may be advantageous to allow winning sets that are subsets of other feasible sets. Clearly in a procurement auction, it is never advantageous to buy more items than needed. If a mechanism were satisfied with buying one item, at the second-lowest price, then buying two items instead would result in paying the third-lowest price for both of them. Hence there is never an advantage to have winning sets that are supersets of other feasible sets.

In some cases we may wish to have only an open set of feasible solutions. By way of example, consider an auction of radio services to subscribers. It may be that, in order to reach some particular set of subscribers, that a particular transmitter would need to be activated — but that this transmitter may also inevitably reach some other set of subscribers, who can then also receive the service.

We will briefly take a look at comparing the expressivity of forward set-system auctions and combinatorial auctions. In order to do this, we will assume that set-systems are closed downwards, and that combinatorial auctions allow free disposal so that the two settings are similar.

We see that the special-case of *single-minded* combinatorial auctions (when each agent values exactly one bundle) are not expressive enough to describe all set-system auctions, and that set-system auctions are trivially not expressive enough to describe all general combinatorial auctions. (It seems likely that set-system auctions lie strictly between single-minded and general combinatorial auctions, in terms of expressivity, but no proof is shown here).

Proposition 6.1. *There exists a forward set-system auction that may not be described by a single-minded combinatorial auction.*

Proof. The proof consists of a short example, as follows

$$\text{Let } \mathcal{E} = \{1, 2, 3\}$$
$$\text{Let } \mathcal{F} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$
$$\text{Let } v_1 = v_2 = v_3 = 1$$

That is, there are three agents who wish to buy some service, but only (any) two may be served simultaneously. In a single-minded combinatorial auction, each agent may value only one bundle of items; hence if two agents share a common item in their bundle, they may not both be selected. Clearly, if this allows the feasible sets given $(\{1,2\},\{1,3\},\{2,3\})$ then none of the agents may share a common item and all three agents could be allocated items simultaneously — giving a possible solution $(\{1,2,3\})$ which is not one of the feasible sets. This set-system auction can, therefore, not be properly described by a single-minded combinatorial auction.

□

We also observe that general combinatorial auctions can express auctions that forward set-system auctions cannot, as follows.

Consider any auction where one agent values two items with different, but non-zero, valuations in a combinatorial auction. A set-system auction allows an agent to be selected, or not, but does not allow a choice of valuations for any agent, so trivially cannot describe a combinatorial auction that allows a choice of valuations.

6.4 Benchmarks

We will now consider the problem of benchmarking forward set-system auctions, in order to be able to provide a reasonable way of measuring the performance of truthful mechanisms, in terms of payments.

6.4.1 Optimal Solution as a Benchmark

As we saw in Chapter 1 for reverse auctions, perhaps the first approach to look for a benchmark would be to take the value of OPT - an optimal solution that may be obtained by an omniscient mechanism,

$$\mathrm{OPT} = \max_{S \in \mathcal{F}} \sum_{e \in S} v_e.$$

However it is well-known, and easy to observe, that attaining even close to an optimal solution may not be possible in the realm of truthful mechanisms. An illustration of this can be seen when only one of the bidders e has a non-zero valuation $v_e > 0$ for some service. An optimal mechanism may be able to offer the service for a fixed price of v_e and agent e may accept, raising revenue v_e. Truthful mechanisms are known to be bid-independent (see, e.g., [16]), hence if agent e wins in a truthful mechanism he will be expected to pay some amount relating to the bids of the competing agents — hence no more than zero. (Observe that this property is satisfied by threshold payments.) Being bid-independent means that the particular payment to some agent must not depend on the value of that agent's bid (or else, given two possible winning bids of b and b' and respective payments p, p' then when $p' > p$ an agent with valuation b may falsely declare b' and strictly increase utility, hence the mechanism could not be truthful). With all other bids being fixed, any variance in one agent's bid may decide if it is chosen in the winning set, or not, but not the payment value that is received if it is chosen. So no truthful mechanism will obtain any revenue for this auction.

As it seems that OPT would be hard to attain, or even approximate, for truthful mechanisms it seems that we need some less demanding benchmark. We will firstly try the relatively naive approach of simply mirroring the NTUmin definition to give F1NTUmax (and observe that $0 \leq$ F1NTUmax $\leq$OPT). We will also try another obvious variant, F2NTUmax, before settling on a third, which we will denote FNTUmax.

6.4.2 Considering F1NTUmax as a Benchmark

As we are looking for a benchmark for selling items, rather than buying, it is reasonable to choose some maximum value rather than the minimum that we used in reverse auctions. Let F1NTUmax (for Forward Non-Transferable Utility max) be defined as follows.

For a given set-system $(\mathcal{E}, \mathcal{F})$, let $n = |\mathcal{E}|$ and suppose there is a valuation vector $\mathbf{v} = (v_1, \ldots, v_n)$. For an instance $I = (\mathcal{E}, \mathcal{F}, \mathbf{v})$ of the problem, define F1NTUmax(I) as

F1NTUmax$(I) = B$ when B is the solution to the following problem.

Let $S \in \text{argmax}_{S \in \mathcal{F}} v_S$ and maximize $B = \sum_{e \in S} b_e$ subject to

(1′) $b_e \leq v_e$ for all $e \in S$

(2′) $\sum_{e \in S \setminus T} b_e \geq \sum_{e \in T \setminus S} v_e$ for all $T \in \mathcal{F}$

(3′) for every $e \in S$, there is $T_e \in \mathcal{F}$ such that $e \notin T_e$ and $\sum_{e' \in S \setminus T_e} b_{e'} = \sum_{e' \in T_e \setminus S} v_{e'}$

Now consider Example 6.1 below as a setting for an auction, which shows F1NTUmax may be arbitrarily lower than the revenue obtained by VCG.

Example 6.1.

Let there be Q identical items to be sold amongst $n = Q + 1$ (unit-demand) bidders. Each competing agent would like to be served, with some valuation for the service, but only Q agents may be served simultaneously. (This is obviously comparable with the single-commodity auctions that were presented in Chapter 3 for the reverse setting.)

Let the valuation vector be

$$\mathbf{v} = (1, \ldots, 1, 0)$$

(i.e., only the final agent n values an item at 0, all other agents value an item at 1). Choosing S optimally gives $S = \{1, \ldots, n-1\}$*, and as* $\max_{e \notin S} v_e = 0$ *in order to satisfy condition (3′) we will have* $b_e = 0$ *for all* $e \in S$ *and hence* F1NTUmax= 0. *Observe that we could use the VCG mechanism for this auction, and could sell Q items with the VCG payment being the same for all agents, the* $(Q+1)$*st highest valuation.*

However, now let us assume that we have an auction where the auctioneer may choose to leave some agents unserved (or some items unsold — this is comparable to the notion of 'free-disposal' in combinatorial auctions). Considering Example 6.1 again, perhaps the mechanism will have chosen, in advance, to implement a k-item Vickrey auction with $k = Q - 1$. A $k-$item Vickrey auction is a single-price auction that sells k items to the k highest bidders at the price equal to the $k+1$st highest bid (see [37]). In this case, each of the $Q-1$ agents chosen would pay a threshold value of 1 (as there is some agent not selected with valuation 1), and the total revenue obtained would be $Q-1$.

It is important to discuss how the mechanism would choose how many items to sell. If the mechanism were to examine the bids first, and then choose k accordingly to optimize revenue then this mechanism would not be truthful.

To illustrate this, consider an example with 2 identical items for sale, to be sold amongst 3 bidders who each value either item equally. Let the valuation vector be $\mathbf{v} = (8, 5, 2)$. If all agents bid truthfully, then the mechanism chooses $k = 1$ to sell 1 item to agent 1 at price $p_1 = 5$, and agent 2 has utility $u_2 = 0$. Alternatively, agent 2 may bid $b_2 = 3$. Now it is optimal for the mechanism to sell 2 items to agents 1 and 2 at price $p_1 = p_2 = 2$; and agent 2 will receive utility $u_2 = v_2 - p_2 = 3$. Therefore, agent 2 has strictly increased utility from 0 to 3 by submitting some bid other than the valuation; which proves that this mechanism is not truthful.

Randomized Mechanisms

If we consider Example 6.1 again, we can propose a randomized mechanism that will choose the number of items k to be sold uniformly at random from $\{1, \ldots, Q\}$ and then proceed with a k-item Vickrey auction. For a randomized mechanism, there is more than one meaning for truthfulness (see,e.g., [2]) — a mechanism that is *truthful in expectation* means that no agent will be benefit, on average, by submitting an untruthful bid, but it is possible that there are occasions when an agent may gain greater utility by submitting an untruthful bid, depending on the choices made by the randomized mechanism. However, there is also a stronger notion of truthfulness — that of being *universally truthful*, which is when, no matter what decisions are made by the randomized algorithm, an agent always maximizes its utility by submitting truthful bids.

Here, we can see that this mechanism is not only truthful in expectation, but also universally truthful, as follows. No bid can affect the choice of k and, given that k has been decided, the k-item Vickrey auction is known to be truthful.

The expected revenue of this mechanism on Example 6.1 is easy to compute, when $k = Q$ then the mechanism obtains revenue 0 and when $k \neq Q$ then the mechanism obtains revenue k. Hence this randomized mechanism will achieve revenue of approximately $Q/2$ in expectation, yet our benchmark figure F1NTUmax would suggest a value of zero, as it has assumed $k = Q$ being the optimal choice of winning set. This suggests that F1NTUmax may be too weak to use as a benchmark, certainly for measuring the payments of randomized mechanisms.

As randomized mechanisms are commonly studied in the literature, (e.g. the Random Sampling Optimal Price (RSOP) auctions and Sampling Cost Sharing (SCS) auctions of Goldberg et al. in [17, 16, 1, 28]) then it would seem prudent to require that any reasonable benchmark should not be arbitrarily smaller than the expected revenue of a truthful randomized mechanism.

Non-Optimal choices of winning set

In the procurement auction the goal of the auctioneer was to minimize the payment. As a consequence of this, it is reasonable to always choose a minimal feasible set — adding superfluous agents to a winning set is undesirable as it will both increase the number of agents that need to be paid and increase the bid values that are considered (as we assume that lowest-bidding agents are more preferred agents and will be chosen first).

However, this coincident behaviour is not maintained when we consider forward auctions. While our auctioneer would like the winning set to be large, to sell as many items as possible, this comes at the cost of decreasing prices (again, we assume that the highest-bidding agents are more preferred and will be chosen first).

As we have seen that truthful mechanisms are bid-independent, we can see that it may certainly be beneficial to sell only a smaller number of items in order to optimize the revenue obtained, as was shown in Example 6.1.

We can observe from condition (3) of the NTUmin definition (Definition 1.2) that the 'bid' value allocated to any agent will be determined by the bids of 'losing' agents. Therefore, when we consider a similar benchmark for forward auctions, we must inevitably conclude that always choosing the revenue-maximizing winning set will give an unreliable benchmark, as it may overly restrict the values of the losing agents. Therefore choosing an optimal winning set may give a value that is arbitrarily lower than some reasonable truthful (possibly randomized) mechanism may be expected to achieve, as can be demonstrated with Example 6.1.

As randomized mechanisms are well-used it seems reasonable to strengthen the benchmark. We may initially attempt this strengthening by giving a choice of winning set rather than only allowing the optimal.

6.4.3 Considering F2NTUmax as a Benchmark

We may first attempt to strengthen the benchmark by considering every possible feasible set as a candidate, and choosing the largest of the resulting values for a benchmark, which we define as F2NTUmax as follows.

F2NTUmax$(I) = \max_{S \in \mathcal{F}} B_S$ when B_S is defined by the solution to the following problem.

maximize $B_S = \sum_{e \in S} b_e$ subject to

$(1'')$ $b_e \leq v_e$ for all $e \in S$

$(2'')$ $\sum_{e \in S \setminus T} b_e \geq \sum_{e \in T \setminus S} v_e$ for all $T \in \mathcal{F}$

$(3'')$ for every $e \in S$, there is $T_e \in \mathcal{F}$ such that $e \notin T_e$ and $\sum_{e' \in S \setminus T_e} b_{e'} = \sum_{e' \in T_e \setminus S} v_{e'}$

However, it quickly becomes obvious that such an approach may lead to set-systems for which this value cannot be defined. For example, let $\mathcal{E} = \{1, 2\}$, let $\mathcal{F} = \{\{1\}, \{2\}\}$, let $v_1 = 1$ and let $v_2 = 0$. If we allowed arbitrary choices of S, we could have $S = \{2\}$ and in order to satisfy condition $(2'')$ (for set $T_2 = \{1\}$) we would have $b_2 \geq v_1$ giving $b_2 \geq 1$. However condition $(1'')$ requires $b_2 \leq 0$, hence the benchmark could not be defined for this set-system as we cannot satisfy the constraints.

In order to avoid this problem, we consider a third approach - by enumerating the proposed benchmark (FNTUmax) over all possible sizes of the winning set, and choosing the largest value. For each possible size, we consider the optimal winning set of that size, and then choose the size, and hence winning set, to be the one that gives the largest value. We can see that, by choosing an optimal winning set for its size, then we do not have the same problem with an undefined benchmark, yet we also get a stronger approach than F1NTUmax.

6.4.4 Considering FNTUmax as a Benchmark

We now define the benchmark FNTUmax as follows;

FNTUmax$(I) = \max_{1 \leq k < n} B_k$ when B_k is defined by the solution to the following problem.

Let $S_k \in \operatorname{argmax}_{S_k \in \mathcal{F}: |S_k| = k} v_{S_k}$ and maximize $B_k = \sum_{e \in S_k} b_e$ subject to

(1^+) $b_e \leq v_e$ for all $e \in S$

(2^+) $\sum_{e \in S \setminus T} b_e \geq \sum_{e \in T \setminus S} v_e$ for all $T \in \mathcal{F}$

(3^+) for every $e \in S$, there is $T_e \in \mathcal{F}$ such that $e \notin T_e$ and $\sum_{e' \in S \setminus T_e} b_{e'} = \sum_{e' \in T_e \setminus S} v_{e'}$

Observe that, in Example 6.1 we have FNTUmax$= Q - 1$ (as $k = Q - 2$ is optimal) and recall that the randomized k-item Vickrey auction suggested gives an expected revenue of approximately $Q/2$, which shows that FNTUmax is at least strong enough to be a reasonable benchmark for this example.

6.4.5 Benchmarks for Unit Demand Auctions

We now consider the special-case of unit-demand combinatorial auctions. These are analogous to the single-commodity auction discussed in Chapter 3, in the forward setting, but further restricted so that each agent only wishes to purchase one item.

In this setting, we have some quantity Q of homogeneous items for sale. There are n agents, each of whom would like to purchase one of these items, and each agent e has some private valuation v_e that he places on receiving the item. Each agent submits one sealed bid b_e and the auctioneer will allocate the Q items (or less), one to each bidder, to give a winning set $S \subseteq \{1, \ldots, n\}$. We assume that each agent $e \in S$ then pays some price $p_e \in [0, b_e]$ for purchasing the item and agents that are not allocated an item do not pay, $\forall e \notin S, p_e = 0$ (i.e., we assume 'no positive transfers' and 'voluntary participation').

This unit-demand auction has been studied previously, notably by Goldberg et al. [16] which examines competitive mechanisms for these goods auction, also with unlimited supply, such as digital goods. In order to establish what they call a competitive framework (this is the same notion of bounding payments that we have referred to as 'frugality') they also require some sort of benchmark figure for their auctions and we reproduce the definition here.

$\mathcal{F}^{(2)}$ is defined as follows: Let $\mathbf{b}$ be a bid vector and let v_e be the e-th largest bid in the vector $\mathbf{b}$. Auction $\mathcal{F}^{(2)}$ on input $\mathbf{b}$ determines the value k such that $k \geq 2$ and kv_k is maximized. All bidders with $b_e \geq v_k$ win at price v_k; all remaining bidders lose. The profit of $\mathcal{F}^{(2)}$ on input $\mathbf{b}$ is thus

$$\mathcal{F}^{(2)}(b) = \max_{2 \leq k \leq n} kv_k$$

We will now see how this benchmark value $F^{(2)}$ is closely related to our FNTUmax, in the special case of unit demand auctions.

Lemma 6.2. *For a unit-demand auction having valuations $v_1 \geq \ldots \geq v_n$, the inequality* FNTUmax $\geq \max_{1 \leq k \leq n} kv_{k+1}$ *holds* .

Proof. Let S^k be the highest-valuation feasible set that contains k agents. For each agent $e \in S^k$, let j be the agent in $\mathcal{E} \setminus S^k$ with the highest valuation. Let $T_e = S^k \setminus \{e\} \cup \{j\}$ be a feasible set (where j replaces e). Condition (2^+) tells us that $b_{S^k \setminus T_e} \geq v_{T_e \setminus S^k}$, which is simplified to $b_e \geq v_j$.

Let $S^k = (1, \ldots, k)$ (recall the agents are sorted into decreasing order of valuation, $v_1 \geq \ldots \geq v_n$), then let $j = k + 1$ and we have $\forall e \leq k, b_e \geq v_{k+1}$.

Therefore, when computing over size k, we have $b_{S^k} \geq kv_{k+1}$. As FNTUmax maximizes over all $1 \leq k \leq n$, then it follows that FNTUmax $\geq \max_{1 \leq k \leq n} kv_{k+1}$,

□

We can now compare this lower bound for FNTUmax with $\mathcal{F}^{(2)}$ and see that the ratio between them is bounded by a factor of 2.

Lemma 6.3. *For unit-demand forward auctions,* $\frac{\mathcal{F}^{(2)}}{\text{FNTUmax}} \leq 2$.

Proof. We have seen the definition that $\mathcal{F}^{(2)}(b) = \max_{2 \leq k \leq n} kv_k$ and Lemma 6.2 which shows FNTUmax $\geq \max_{1 \leq k \leq n} kv_{k+1}$. Fixing the k which maximizes $\mathcal{F}^{(2)}$, let $k' = k - 1$. We saw in the proof of Lemma 6.2 that $b_{S^{k'}} \geq k'v_{k'+1}$ and hence FNTUmax $\geq k'v_{k'+1}$. We can rewrite this (substituting $k - 1 = k'$) as

$$\text{FNTUmax} \geq (k-1)v_k,$$

and, as we have fixed k to be the maximum, from its definition we have

$$\mathcal{F}^{(2)} = kv_k.$$

Hence doing the division gives

$$\frac{\mathcal{F}^{(2)}}{\text{FNTUmax}} \leq \frac{k}{k-1},$$

and as $k \geq 2$, from the definition of $\mathcal{F}^{(2)}$, this can be simplified to

$$\frac{\mathcal{F}^{(2)}}{\text{FNTUmax}} \leq \frac{2}{1}$$

□

Now Lemma 6.3 allows us to leverage the results from [16], to show that no deterministic mechanism can be competitive with respect to FNTUmax. Their theorem statement is reproduced here.

> *Theorem 4.1 Let $\mathcal{A}_f$ be any symmetric deterministic auction defined by bid-independent function f. Then $\mathcal{A}_f$ is not competitive: For any $1 \leq m \leq n$ there exists a bid vector* **b** *of length n such that the profit of $\mathcal{A}_f$ on* **b** *is at most* $\mathcal{F}^{(m)}(\mathbf{b})\frac{m}{n}$

Using Lemma 6.3 we can fix $m = 2$ to give a similar result for FNTUmax.

Corollary 6.4. *Let $\mathcal{A}_f$ be any symmetric deterministic auction defined by bid-independent function f. Then there exists a bid vector* **b** *of length n such that the profit of $\mathcal{A}_f$ on* **b** *is at most* FNTUmax$(\mathbf{b})\frac{4}{n}$.

6.4.6 Considering Alternatives to FNTUmax

As we have noted that NTUmax can be used as an alternative benchmark to NTUmin, and is in some ways more desirable, we could consider using FNTUmin rather than FNTUmax. If we minimize over the choice of the size parameter, k, then the auction given in Example 6.1 (with valuations $(1, 1, \epsilon, \ldots, \epsilon, 0)$) will result in a benchmark value of 0, which is unrealistically low (by choosing $k = Q$). Hence we must still maximize over the size parameter k, yet we will minimize the value given by a fixed winning set, as follows.

Define FNTUmin(I) $= \max_{1\leq k<n} B_k$ when B_k is defined by the solution to the following problem.

Let $S_k \in \mathrm{argmax}_{S_k \in \mathcal{F}:|S_k|=k} v_{S_k}$ and minimize $B_k = \sum_{e\in S_k} b_e$ subject to (1^+),(2^+) and (3^+).

It is worth observing that in the case of the unit-demand auctions discussed earlier, that we will always have FNTUmin = FNTUmax; the bid of each agent $e \in S$ is defined by the feasible set $T_e = S \setminus \{e\} \cup \{j\}$ (i.e., $b_e = v_j$) when j is the item in $\mathcal{E} \setminus S$ with the largest valuation. This means that each bid is defined by the valuation of a single agent and there is no variation in the bids, hence the minimum and maximum values are equal.

Chapter 7

Conclusion and Discussion

7.1 Conclusion

To conclude, for each of the chapters, we will examine the main results, give a discussion on the impact of these results and consider the main questions that have been left open by this work.

7.1.1 Discussion and Summary of Main Results

Chapter 2

Chapter 2 gives a result for the frugality of VCG that seems to be reasonably obvious, but not specifically documented elsewhere. The result for the frugality of monotonic approximation mechanisms, in general, is an extension of this. Again, this result does not appear to be documented elsewhere and is a more generalized version of Theorem 18 in [11]. Showing that only minimal winning sets need be considered with regard to tie-breaking (for VCG) is, likewise, fairly obvious but has been documented for completeness.

Chapter 3

We introduced a very natural single-commodity auction. We show that even in a very limited special case where only $\{1, 2\}$ quantities are permitted, VCG has poor frugality (with respect to NTUmin). We gave a mechanism that greatly improves the frugality in this special case. This result is within a constant factor of optimal for similar types of mechanism, but we have not shown that it is close to optimal for all truthful mechanisms (although it does seem likely). We have also shown a lower bound for the general case which shows that a blind-scaling mechanism of the same type will only be able to achieve relatively small gains in frugality. We conjecture that there is some scaling mechanism that may gain an improvement in frugality, by preferring agents with larger quantities over smaller ones, but have been unable to provide any proof, and confirming this remains an open question.

One obvious open question for both these cases is whether the scaling mechanism could do better. There may be other types of (non-linear) scaling mechanism that have better frugality, which has not been addressed here. Perhaps allowing a mechanism more information about the instance, before it decides on scaling (or other) factors, would help. If more information helps lower frugality, then the question arises of exactly which information about the instance a mechanism would need in order to improve frugality, yet could still do so truthfully. Rather than the lower bound we have, restricted to a class of 'blind-scaling' mechanisms, one goal would be to find a lower bound on frugality for all truthful mechanisms.

We could extend the scope of this chapter to relax the requirement that quantities are integer, and examine the frugality of the mechanisms in that setting. It seems unlikely that a simple blind-scaling mechanism will achieve better frugality than VCG, as the range of quantities in the instances can vary arbitrarily.

Chapter 4

The generalization of path-auctions given in Chapter 4, which assumes that agents may sell bundles of edges, appears to have some reasonable motivation, although we show that finding an exact solution is NP-hard. The polynomial-time mechanism shown for it has an unsatisfactorily large frugality ratio (due to only a naive approximation algorithm being used). The best approximation ratio we have is k (the number of edges owned by each agent), but we do not have an inapproximability result showing that it would be hard to do better, so this remains an obvious open question. Scaling approaches for path-auctions have been shown to work well (e.g. [24, 39], and we have seen mechanisms proposed recently ([5] and [25]) that give a better frugality ratio than VCG for some problems, mostly related to vertex covers.

It is a natural question to ask if similar approaches could improve frugality for this auction (even for the NP-hard exact solution). An interesting direction for future research would be to look for a truthful mechanism that can implement a better approximation algorithm (if it can be shown that one exists) and that can also improve frugality by scaling. By doing this, it may be possible to create a genuinely practical auction — that is both tractable and has good frugality.

Chapter 5

Chapter 5 examines a number of methods of obtaining prices through first-price auctions which could be considered as possible alternatives to NTUmin as benchmarks for measuring frugality. We initially described a way to look at finding feasible first-price bids on set-system auctions as a hypergraph of constraint sets.

Recently the focus has been more on using NTUmax as a benchmark rather than NTUmin (e.g. [5, 25]). As we show that NTUmin may be hard to approximate, this is a reason for intuitively believing that it may be unrealistically low for use as a benchmark.

The uniformly rising first-price auction that was proposed seems to be a good way of producing a reasonable single solution but it may prove to be difficult to analyse.

We considered the process of taking maximal bids from agents in some order and saw that this ordering always produces a range of values between NTUmin($\mathbf{c}$) and NTUmax($\mathbf{c}$) and that certain instances allow both extremes to be reached with the appropriate ordering. This is another good way to produce some feasible solution (e.g. with a random ordering), but this still possibly leaves a large range of values to choose from and finding the ordering that gives a minimum is hard to approximate. We also showed that NTUmin may be hard to approximate even where calculating NTUmax is tractable (such as for edge cover auctions).

In the general case, even with just ternary constraint sets, we have seen examples that the minimum ordering value, OMBmin($\mathbf{c}$), may be significantly higher than NTUmin($\mathbf{c}$) (i.e., OMBmin($\mathbf{c}$) $\geq 2n$(NTUmin($\mathbf{c}$)/9), as well as an example where OMBmax($\mathbf{c}$) may be lower than NTUmax($\mathbf{c}$). Where the size of the winning set S is restricted to $|S| \leq 4$, we saw that OMBmin($\mathbf{c}$) = NTUmin($\mathbf{c}$), but that this does not hold where $|S| \geq 5$.

In the restricted case of binary-single value constraints, we saw a proof that that OMBmin($\mathbf{c}$) $\leq$ NTUmin($\mathbf{c}$)/2 and a class of examples for which OMBmin($\mathbf{c}$) / NTUmin($\mathbf{c}$) approaches 2, matching the lower bound. Even in this restricted setting, we saw that approximating either NTUmin($\mathbf{c}$) or OMBmin($\mathbf{c}$) to within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$ is NP-hard.

In the case of single-commodity auctions we saw that OMBmin($\mathbf{c}$) may be greater than NTUmin($\mathbf{c}$) by up to a factor of 2, but we do not currently know of any upper bound for this ratio (other than the trivial $|S|$). It remains an open question whether it would always be possible to find an ordering σ such that $b_S^\sigma \leq 2$NTUmin($\mathbf{c}$). However, we have seen that a number of reasonable heuristics for creating an ordering σ give $b_S^\sigma > 2$ NTUmin($\mathbf{c}$) for some examples.

Chapter 6

Chapter 6 considered possible benchmark values for forward auctions. We saw that the definitions used in the reverse auctions can not immediately be adapted, but that by expanding the scope we can give a possible candidate, which at least seems to be adequate for unit-demand auctions.

We saw that the problems of defining a benchmark for forward and reverse auctions are fundamentally quite different and that in the forward setting we cannot reasonably hope to use a benchmark that is simply based upon choosing a single, optimal, feasible set, as was the case for reverse auctions.

We have proposed an auction framework for forward auctions based on the set-system idea used in procurement auctions. While we did not give many details of the expressivity of this auction, it seems likely that it lies strictly between single-minded combinatorial auctions and general combinatorial auctions, so finding an exact characterization may be an interesting problem left open.

We have proposed a benchmark, FNTUmax, based on the definitions used for NTU-min in procurement auctions, that can be applied to all forward set-system auctions. As this involves computing over all possible sizes of feasible sets and not just minimal feasible sets as in procurement auctions, it is likely to be harder to analyse. However, we have seen that in the special case of unit-demand auctions, it is close to a previously-used benchmark, $\mathcal{F}^{(2)}$, which suggests that it may have some potential.

It remains to be seen if this FNTUmax value could be used as a benchmark figure for more expressive auctions, such as general combinatorial auctions. Due to the nature of counting over all sizes of feasible sets, it is possible that this will prove too strong a benchmark such that not even randomized mechanisms will be able to give some performance guarantee with respect to FNTUmax, this is an area that has not yet been addressed.

Bibliography

[1] Alaei, S., Malekian, A., Srinivasan, A.: On random sampling auctions for digital goods. In: Proceedings of the 10th ACM conference on Electronic commerce. pp. 187–196 (2009)

[2] Archer, A., Papadimitriou, C., Talwar, K., Tardos, E.: An approximate truthful mechanism for combinatorial auctions with single parameter agents. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 205–214 (2003)

[3] Archer, A., Tardos, E.: Truthful mechanisms for one-parameter agents. In: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science. pp. 482– (2001)

[4] Archer, A., Tardos, E.: Frugal path mechanisms. In: ACM Trans. Algorithms. vol. 3, pp. 3:1–3:22 (Feb 2007)

[5] Chen, N., Elkind, E., Gravin, N., Petrov, F.: Frugal mechanism design via spectral techniques. In: Foundations of Computer Science, Annual IEEE Symposium on. pp. 755–764 (2010)

[6] Clarke, E.H.: Multipart pricing of public goods. In: Public Choice. vol. 11 (September 1971)

[7] Damian-Iordache, M., Pemmaraju, S.V.: Hardness of approximating independent domination in circle graphs. In: Proceedings of the 10th International Symposium on Algorithms and Computation. pp. 56–69 (1999)

[8] Demange, G., Gale, D., M.Sotomayor: Multi-item auctions. In: Journal of Political Economy. vol. 94, pp. 863–872 (1986)

[9] Dijkstra, E.W.: A note on two problems in connexion with graphs. In: Numerische Mathematik. vol. 1, pp. 269–271 (1959)

[10] Du, Y., Sami, R., Shi, Y.: Path auctions with multiple edge ownership. In: Theoretical Computer Science. vol. 411, pp. 293 – 300 (2010)

[11] Elkind, E., Goldberg, L., Goldberg, P.: Frugality ratios and improved truthful mechanisms for vertex cover. In: Proceedings of the 8th ACM Conference on Electronic Commerce. pp. 336–345 (2007)

[12] Elkind, E., Sahai, A., Steiglitz, K.: Frugality in path auctions. In: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 701–709 (2004)

[13] Feige, U.: A threshold of $\ln n$ for approximating set cover. In: Journal of the ACM. vol. 45, pp. 314–318 (1998)

[14] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness (1979)

[15] Garfinkel, R.S., Nemhauser, G.L.: Integer programming. Wiley-Interscience (1972)

[16] Goldberg, A.V., Hartline, J.D., Karlin, A.R., Saks, M., Wright, A.: Competitive auctions. In: Games and Economic Behavior. vol. 55, pp. 242 – 269 (2006), mini Special Issue: Electronic Market Design

[17] Goldberg, A.V., Hartline, J.D., Wright, A.: Competitive auctions and digital goods. In: Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 735–744 (2001)

[18] Groves, T.: Incentives in teams. In: Econometrica. vol. 41, pp. 617–31 (July 1973)

[19] Halldórsson, M.M.: Approximating the minimum maximal independence number. In: Information Processing Letters. vol. 46, pp. 169 – 172 (1993)

[20] Hassin, R., Levin, A.: A better-than-greedy approximation algorithm for the minimum set cover problem. In: SIAM J. Comput. vol. 35, pp. 189–200 (Jul 2005)

[21] Herodotus, Dewald, C., Waterfield, R.: The Histories. Oxford Paperbacks (2008)

[22] Immorlica, N., Nikolova, E., Karger, D., Sami, R.: First-price path auctions. In: Proc. 7th ACM Conf. on Electronic Commerce. pp. 203–212 (2005)

[23] Iwasaki, A., Kempe, D., Saito, Y., Salek, M., Yokoo, M.: False-name-proof mechanisms for hiring a team. In: Proceedings of the 3rd international conference on Internet and network economics. pp. 245–256 (2007)

[24] Karlin, A.R., Kempe, D., Tamir, T.: Beyond VCG: Frugality of truthful mechanisms. In: FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science. pp. 615-626 (2005)

[25] Kempe, D., Salek, M., Moore, C.: Frugal and truthful auctions for vertex covers, flows and cuts. In: Foundations of Computer Science, Annual IEEE Symposium on. pp. 745–754 (2010)

[26] Khachiyan, L.: A polynomial algorithm in linear programming. Soviet Mathematics Doklady 20, 191–194 (1979), (English Translation)

[27] Korte, B.H.: Combinatorial Optimization : Theory and Algorithms. Springer-Verlag (2006)

[28] Koutsoupias, E., Pierrakos, G.: On the competitive ratio of online sampling auctions. In: Proceedings of the 6th international conference on Internet and network economics. pp. 327–338 (2010)

[29] Lusht, K.M.: A comparison of prices brought by English auctions and private negotiations. In: Real Estate Economics. vol. 24, pp. 517–530 (1996)

[30] Southern Railway: DaySave Information (2012 – accessed May 31, 2012), `http://www.southernrailwaytickets.com/daysave.php`

[31] Merseytravel: Saveaway Ticket Information (2012 – accessed May 31, 2012), `http://www.merseytravel.gov.uk/information_tickets_saveaway.asp`

[32] Murty, K.G., Perin, C.: A 1-matching blossom-type algorithm for edge covering problems. In: Networks. vol. 12, pp. 379–391 (1982)

[33] Nisan, N., Ronen, A.: Algorithmic mechanism design (extended abstract). In: Proceedings of the Thirty-first Annual ACM Symposium on Theory Of Computing. pp. 129–140 (1999)

[34] Nisan, N., Ronen, A.: Computationally feasible VCG mechanisms. In: Proceedings of the 2nd ACM conference on Electronic commerce. pp. 242–252 (2000)

[35] Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press (2007)

[36] Talwar, K.: The price of truth: Frugality in truthful mechanisms. In: Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science. pp. 608–619 (2003)

[37] Vickrey, W.: Counterspeculation, Auctions, and Competitive Sealed Tenders. In: The Journal of Finance. vol. 16, pp. 8–37 (1961)

[38] de Vries, S., Vohra, R.V.: Combinatorial auctions: A survey. In: INFORMS Journal on Computing. pp. 284–309. No. 3 (2003)

[39] Yan, Q.: On the price of truthfulness in path auctions. In: Proceedings of the 3rd international conference on Internet and network economics. pp. 584–589 (2007)